XIII-th International Conference

Knowledge-Dialogue-Solution

June 18-24, 2007, Varna (Bulgaria)

P R O C E E D I N G S

Volume 2

ITHEA

SOFIA, 2007

Gladun V.P., Kr.K. Markov, A.F. Voloshin, Kr.M. Ivanova (editors)

Proceedings of the XIII-th International Conference "Knowledge-Dialogue-Solution" –

Varna, 2007

Volume 2

Sofia, Institute of Information Theories and Applications FOI ITHEA, Bulgaria– 2007

First Edition

The XIII-th International Conference "Knowledge-Dialogue-Solution" (KDS 2007) continues the series of annual international KDS events organized by Association of Developers and Users of Intelligent Systems (ADUIS).

The conference is traditionally devoted to discussion of current research and applications regarding three basic directions of intelligent systems development: knowledge processing, natural language interface, and decision making.

Edited by:

Institute of Information Theories and Applications FOI ITHEA, Bulgaria

Association of Developers and Users of Intelligent Systems, Ukraine

Printed in Bulgaria by Institute of Information Theories and Applications FOI ITHEA

Sofia-1090, P.O.Box 775, Bulgaria

e-mail: info@foibg.com

www.foibg.com

ISSN 1313-0087 (paperback) ISSN 1313-1176 (CD) ISSN 1313-1206 (online)

FSBN Volume 1: 978-954-16-2007-6 (paperback) 978-954-16-2008-3 (CD) 978-954-16-2009-0 (Online)

FSBN Volume 2: 978-954-16-2010-6 (paperback) 978-954-16-2011-3 (CD) 978-954-16-2012-0 (Online)

PREFACE

The scientific XIIIth International Conference “Knowledge-Dialogue-Solution” took place in June, 18-24, 2007 in Varna, Bulgaria. This two volumes include the papers presented at the conference. Reports contained in the Proceedings correspond to the scientific trends, which are reflected in the Conference name.

The Conference continues the series of international scientific meetings, which were initiated more than fifteen years ago. It is organized owing to initiative of ADUIS - Association of Developers and Users of Intelligent Systems (Ukraine), Institute of Information Theories and Applications FOI ITHEA, (Bulgaria), and IJ ITA - International Journal on Information Theories and Applications, which have long-term experience of collaboration.

Now we can affirm that the international conferences “Knowledge-Dialogue-Solution” in a great degree contributed to preservation and development of the scientific potential in the East Europe.

KDS-2007 is dedicated to:

- 60th Anniversary of the Institute of Mathematics and Informatics of Bulgarian Academy of Sciences;
- 15th Anniversary of the Association of Developers and Users of Intelligent Systems (Ukraine);
- 10th Anniversary of the Association for Development of the Information Society (Bulgaria).

The conference is traditionally devoted to discussion of current research and applications regarding three basic directions of intelligent systems development: knowledge processing, natural language interface, and decision making.

The basic approach, which characterizes presented investigations, consists in the preferential use of logical and linguistic models. This is one of the main approaches uniting investigations in Artificial Intelligence.

The organization of the papers in KDS-2007 is based on specialized sessions. They are:

Neural Nets
Mathematics of Computing
Decision Tools and Techniques
Decision Support
Formal Models
Expert Systems
Ontologies
Knowledge Acquisition
Natural Language Processing
Pattern Recognition
Distributed Information Processing
Philosophy and Methodology of Informatics

The official languages of the Conference are Russian and English.

The Conference is sponsored by **FOI Bulgaria** (www.foibg.com).

We appreciate the contribution of the members of the KDS 2007 Program Committee.

On behalf of all the conference participants we would like to express our sincere thanks to everybody who helped to make conference success and especially to Kr.Ivanova, I.Mitov, V.Velichko and L. Svyatogor.

Chairmen of the Program Committee: Victor Gladun, Alexey Voloshin, Krassimir Markov

CONFERENCE ORGANIZERS

National Academy of Sciences of Ukraine

Association of Developers and Users of Intelligent Systems (Ukraine)

International Journal “Information Theories and Applications”

V.M.Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine

Institute of Information Theories and Applications FOI ITHEA (Bulgaria)

Institute of Mathematics and Informatics, BAS (Bulgaria)

Institute of Mathematics of SD RAN (Russia)

PROGRAM COMMITTEE

Victor Gladun (Ukraine)

Alexey Voloshin (Ukraine)

Krassimir Markov (Bulgaria)

Igor	Arefiev (Russia)	Genady	Osipov (Russia)
Frank	Brown (USA)	Alexander	Palagin (Ukraine)
Vladimir	Donskoy (Ukraine)	Vladimir	Pasechnik (Ukraine)
Alexander	Eremeev (Russia)	Zinoviy	Rabinovich (Ukraine)
Natalia	Filatova (Russia)	Alexander	Reznik (Ukraine)
Constantine	Gaindric (Moldova)	Vladimir	Ryazanov (Russia)
Tatyana	Gavrilova (Russia)	Galina	Rybina (Russia)
Krassimira	Ivanova (Bulgaria)	Vasil	Sgurev (Bulgaria)
Vladimir	Jotsov (Bulgaria)	Vladislav	Shelepov (Ukraine)
Julia	Kapitonova (Ukraine)	Anatoly	Shevchenko (Ukraine)
Vladimir	Khoroshevsky (Russia)	Ekaterina	Solovyova (Ukraine)
Rumyana	Kirkova (Bulgaria)	Vadim	Stefanuk (Russia)
Nadezhda	Kiselyova (Russia)	Tatyana	Taran (Ukraine)
Alexander	Kleshchev (Russia)	Valery	Tarasov (Russia)
Valery	Koval	Adil	Timofeev (Russia)
Oleg	Kuznetsov (Russia)	Vadim	Vagin (Russia)
Vladimir	Lovitskii (GB)	Jury	Valkman (Ukraine)
Vitaliy	Lozovskiy (Ukraine)	Neonila	Vashchenko (Ukraine)
Nadezhda	Mishchenko (Ukraine)	Stanislav	Wrycza (Poland)
Ilia	Mitov (Bulgaria)	Nikolay	Zagoruiko (Russia)
Xenia	Naidenova (Russia)	Jury	Zaichenko (Ukraine)
Olga	Nevzorova (Russia)	Arkady	Zakrevskij (Belarus)

Official languages of the conference are Russian and English.

General sponsor of the KDS 2007 is **FOI BULGARIA** (www.foibg.com).

TABLE OF CONTENTS – VOLUME 1

D.2.2. Decision Tools and Techniques

H.4.2. Decision Support

E.4 Formal Models

About:

TABLE OF CONTENTS – VOLUME 2

I.2.6. Knowledge Acquisition

I.2.7. Natural Language Processing

I.5. Pattern Recognition

C.2.4. Distributed Information Processing

I.2.0. Philosophy and Methodology of Informatics

About:

INDEX OF AUTHORS

I.2.1. Expert Systems

THE EXPERIENCE OF DEVELOPMENT AND APPLICATION PERSPECTIVES OF LEARNING INTEGRATED EXPERT SYSTEMS IN THE EDUCATIONAL PROCESS

Galina Rybina, Victor Rybin

***Abstract**: The main principles and experience of development of learning integrated expert systems based on the third generation instrumental complex AT-TECHNOLOGY are considered.*

***Keywords**: learning integrated expert systems, problem-oriented methodology, educational process.*

***ACM Classification Keywords**: I.2..1 Artificial Intelligence: Applications and Expert Systems*

The most important property of next generation intelligent learning systems is the possibility of *individualization* for learning processes both with the help of using different remote controlled educational technologies and further integration of models, methods and technologies related to expert systems with learning systems in the context of united architecture of integrated expert system (IES) which combine interacted logical-linguistic, mathematical, imitating and some other kinds of models.

The problem-oriented methodology of construction IES that was offered in the middle of nineties [Rybina, 1997] and unique next generation tooling supported it - the complex AT-TECHNOLOGY [Rybina, 2005, Rybina, 2004] allow to realize the development including wide class *of learning* IES having advanced intelligent resources in leaning, monitoring and testing of the trainee that suppose:

- construction of the *trainee's model* (considered personal psychological portrait) and *sample model* of a course (in particular developing before *teacher's model);*
- construction of adaptive *learning model* which essence is in dynamic modification of *learning strategy* in compliance with current trainee's model and following generation of the set of *teaching actions* most effective on the given learning step considered psychological features of trainees;
- trainee's activity control and generation of controlling decisions for the corresponding adjustment of trainee's activity with the purpose of achievement of given educational goals;
- construction of the model of problem domain and explanation model for the assessment of the decision-making logic, calculation results, explanation (if necessary) for wrong alternative or problem-solving step;
- possibility of using hypertext internet-textbook, playing programs, etc, having standard state of teaching actions.

All models, techniques, algorithms and procedures formed in aggregate a concrete methodology of construction of learning IES in the contest of problem-oriented methodology of construction of wide IES class should be noted as original (published in 38 papers); and supported instrumental tools embedded in the complex AT-TECHNOLOGY present itself automated workplace for subject-teachers in engineering and specialized

disciplines, i.e. those disciplines which are expedient for creating learning IES like training simulators of teaching kind with the purpose of saving of the unique non-formalized techniques and experience of concrete courses and disciplines teaching.

The experience of using several generations of complex AT-TECHNOLOGY for development of a number of learning IES also showed great perspectives for the creation of web-oriented IES just for educational purposes since, on the one hand, powerful functionality of the learning IES (the construction of trainee's model, adapted model of learning, model of problem domain, explanation model, teacher's model) is wholly inherited, on another hand, all basic features of contemporary client-server architecture such as system independence from platform, accessibility, simplicity of informational renewal, convenience in administration and technical support that in particular simplify processes of subject-teachers knowledge accumulation noticeably.

Experimental approbation of dynamically developing supportive tools for construction of learning IES functioned in compound of third generation complex AT-TECHNOLOGY were held on the example of development:

- learning IES on the courses "Designing systems based on knowledge" and "Intellectual dialogue systems" (department of Cybernetics of Moscow Engineering Physics Institute (State University) – (MEPhI);
- learning IES on the course "Automation of experimental physical devices" (department of Automatics of MEPhI);
- learning IES on the differential diagnostics of insult kinds (together with Scientific Research Institute of Neurology Russian Academy of Sciences);
- learning IES for the diagnostics of respiratory tract illnesses (together with children's municipal polyclinic № 109 North-West Administrative District Moscow), which demonstration is provided for the exhibition "Telecommunications and new informational technologies in education".

As a whole complex AT-TECHNOLOGY is a multifunctional automatic workplace for knowledge engineers and also students and post-graduate students studying the theory and technology of construction IES that since 1995 allowed complex to use efficiently in educational process in MEPhI and other institutes for specialists preparation in the area of static and dynamic IES and knowledge management systems also [Rybina, 2005].

As a basic software tool complex AT-TECHNOLOGY is included in the structure of imitating-simulated stand (IMS) constructed in the educational-scientific laboratory "Systems of Artificial Intelligence" department of Cybernetics MEPhI on the base of local web consisting of 8 PC Pentium connecting to Internet-web MEPhI. In the compound of software tools of IMS there are foreign licensed products G2, GDA, Telewindows, etc. that are used for practical support of the courses and disciplines in departments of Cybernetics, System Analisys, Automatics [Rybina et.al., 2004, Koltsov et.al., 2006].

Acknowledgements

This work was supported by the Russia Foundation for Basic Research project no 06-01-00242

Bibliography

[Rybina, 1997] G.V Rybina. Problem-oriented methodology of automatic construction of integrated expert systems for static problem domains. In: Proceedings of the Russian Academy of Sciences. Theory and management systems. 1997. №5. P. 129-137.

[Rybina, 2005] G.V Rybina. Automatic workplace for construction of integrated expert systems: complex AT-TECHNOLOGY. In: Artificial Intelligence news. 2005. №3. P. 69-87.

[Rybina, 2004] G. V. Rybina. Instrumental tools of next generation for the construction of applied instrumental systems. In: Aero-space instrument-making industry. 2004. №10. P. 14-23.

[Rybina, 2005] G.V. Rybina. Instrumental base for the preparation of specialists in the area of intelligent systems and technologies. In: International scientific-practical conference "Reengineering of business processes based on contemporary informational technologies. Knowledge management systems. ": Collection of scientific papers. M.: MESI, 2005.

[Rybina et.al., 2004] G. V. Rybina, V.Yu. Berzin. Laboratory practical work on the course Dynamic intellectual systems": Textbook M.: MEPhI, 2004. 96p.

[Koltsov et.al., 2006] I.M Koltsov, A.V. Konovalov, D.E. Manuhin, A.V. Pchelintsev, V.M. Rybin. Contemporary technologies of automatics. Textbook. M.: MEPhI, 2006. 92 p.

Authors' Information

Galina Rybina *– Moscow Engineering Physics Institute (State University), Kashirskoe shosse, 31, 115049, Moscow, Russia, e-mail: galina@ailab.mephi.ru*

Victor Rybin *– Engineering Physics Institute (State University), Kashirskoe shosse, 31, 115049, Moscow, Russia, e-mail: VMRybin@mephi.ru*

КОНЦЕПЦИЯ КОМПЬЮТЕРНОГО БАНКА ЗНАНИЙ ПО МЕДИЦИНСКОЙ ДИАГНОСТИКЕ

Филипп Москаленко, Александр Клещёв, Мери Черняховская

Аннотация*: В работе описано информационное и программное наполнение компьютерного банка знаний по медицинской диагностике. Определены классы его пользователей и задачи, ими решаемые. Информационное наполнение специализированного банка знаний содержит три онтологии: онтологию наблюдений в области медицинской диагностики, онтологию базы знаний (заболеваний) по медицинской диагностике и онтологию историй болезни, а также три класса информационных ресурсов в различных областях медицины – базы наблюдений, базы знаний, и базы данных (пациентов), соответствующие этим онтологиям. Программное наполнение содержит редакторы информации (онтологий, баз наблюдений, знаний и данных), а также решатель задачи медицинской диагностики.*

Keywords*: Medical Diagnostics, ontology model, parallel computing, knowledge bank.*

ACM Classification Keywords*: I.2.1 Applications and Expert Systems, J.3 Life and Medical Sciences.*

Введение

Одним из приложений систем искусственного интеллекта являются системы медицинской диагностики. Их применение помогает врачу повысить качество своей работы. Задачей таких систем является определение заболеваний (одного или нескольких), которыми возможно болен пациент, на основе данных о его наблюдениях. Необходимыми компонентами таких систем являются подсистема доверия, которая показывает пользователям-врачам, какими знаниями обладает система диагностики, и подсистема объяснения, которая разъясняет пользователю, на основе каких рассуждений и знаний системой предлагаются те или иные решения.

В настоящее время для решения задачи медицинской диагностики разрабатываются два класса систем, различающиеся методами, которые положены в их основу. Один класс составляют системы, основанные на статистических и других математических моделях – их основой служат математические алгоритмы, занимающиеся поиском обычно частичного соответствия между симптомами очередного пациента и симптомами наблюдавшихся ранее пациентов, диагнозы которых известны [1 – 4]. Однако такие системы

не имеют подсистем доверия и средств формирования понятного врачам объяснения полученного результата.

Системы второго класса основаны на знаниях экспертов. В них алгоритмы оперируют информацией о пациенте и знаниями о заболеваниях, представленными в форме, в той или иной степени приближенной к представлениям врачей (и описанных экспертами-врачами), что достигается за счёт явного или неявного использования онтологий медицинской диагностики. Именно в системах такого типа возможно создание подсистем доверия, а также построение компонента объяснения, способного дать врачу результаты анализа данных пациента, которые привели к полученному решению задачи.

Используемые в таких системах модели онтологии учитывают изменение значений признаков во времени [1], связи симптомов и заболеваний (например, с помощью логических правил) [2, 5], деление наблюдений на несколько групп (например, клинические, лабораторные, морфологические данные в [4]), представление состояния пациента в виде многоуровневой модели [6].

Алгоритмы в таких системах пытаются имитировать ход рассуждений врача при постановке диагноза [7, 8], занимаются поиском соответствия информации о больном и клинических картин заболеваний, описанных врачом [9], либо обработкой заданных экспертами правил, описывающих связи наблюдений и заболеваний [5, 8, 10, 11].

Анализ разрабатываемых в последние годы систем медицинской диагностики второго класса показал, что используемые в них онтологии медицинской диагностики являются сравнительно простыми и одновременно не отражают такие повсеместно используемые врачами в их практике знания предметной области как: знания о причинах заболеваний; знания о различных типах причинных связей между признаками и заболеваниями; знания о воздействиях событий на значения признаков при заболеваниях и у здоровых пациентов; знания о различных вариантах изменения значений признаков, зависящие от анатомо-физиологических особенностей пациентов.

Помимо этого, одним из отрицательных свойств некоторых разрабатываемых систем является то, что круг их применения достаточно узок. Это обусловлено тем, что они представляют собой либо макетные версии, выполненные для каких-либо исследовательских целей, либо разработаны для определённого медицинского учреждения и не доступны за рамками его локальной сети. С другой стороны, системы медицинской диагностики, предоставляющие широкий доступ к своим ресурсам с применением современных сетевых технологий (Интернет), например DXplain [2] и Диагностика преэклампсии [12], не позволяют экспертам расширять используемые в них базы знаний.

Таким образом, актуальной задачей является разработка системы медицинской диагностики, основанной на знаниях экспертов и модели онтологии, учитывающей все приведённые выше особенности медицинских знаний, в которой их модель имеет форму наиболее близкую к представлениям экспертов и позволяет определять не только диагноз пациента, но и объяснять его. Такая система должна проводить диагностику за приемлемое для врача время (несмотря на то, что в её основе лежит нетривиальная онтология медицинских знаний). Кроме того, такая система должна предоставлять доступ как можно большему числу пользователей, как для проведения медицинской диагностики, так и для участия в накоплении и совершенствовании медицинских знаний о различных заболеваниях.

Цель данной работы – описание концепции сетевого ресурса по медицинской диагностике, обладающего указанными выше свойствами.

Работа выполнена в рамках конкурсного проекта ДВО РАН № 06-III-А-01-457 «Проектирование, разработка и развитие банка медицинских знаний в сети Интернет».

1. Теоретические предпосылки и общие принципы создания банка знаний по медицинской диагностике

За последние годы специалистами в области искусственного интеллекта и экспертами в области медицинской диагностики были разработаны несколько моделей онтологии медицинской диагностики.

Некоторые их этих моделей применялись при разработке систем диагностики, основанных на знаниях экспертов. Как отмечалось во введении, каждая из них в чём-то была лучше, а в чём-то хуже других [1, 2, 4, 5, 6]. Для объединения преимуществ этих моделей и создания модели онтологии, наиболее приближенной к представлениям знаний в области медицинской диагностики была разработана и описана онтология и её модель [13,14,15]. Данная онтология медицинской диагностики описывает острые заболевания, в ней учитывается взаимодействие причинно-следственных отношений различных типов. Она близка к реальным представлениям медицины в Российской Федерации и описывает сочетанную и осложненную патологии, динамику патологических процессов во времени, а также воздействие лечебных мероприятий и других событий на проявления заболеваний. Моделью этой онтологии является необогащённая система логических соотношений с параметрами, которая состоит из определений терминов модели действительности (неизвестных), определений терминов модели знаний (параметров), ограничений целостности неизвестных и параметров, а также соотношений между ними.

Соотношения между неизвестными и параметрами делятся на следующие смысловые группы:

1) Соотношения между знаниями о причинно-следственных отношениях и причинно-следственными связями, имеющими место в ситуации.

2) Соотношения, определяющие причинно-следственные связи, являющиеся причинами значений каждого признака на разных интервалах времени.

3) Соотношения, определяющие для каждого признака свойства границ интервалов разбиения оси времени, связанного с этим признаком.

4) Соотношения, определяющие причину для каждого заболевания, входящего в диагноз.

На основе данной модели онтологии в работе [16] поставлена общая задача медицинской диагностики: определить возможные диагнозы больного на основе знаний предметной области и данных его обследования, к которым относятся значения признаков (в моменты их наблюдения), значения его анатомо-физиологических особенностей (постоянные во времени) и значения произошедших с ним событий (в моменты, когда они происходили), а также для каждого такого диагноза сформировать его причину (некоторое событие или другие заболевание) и объяснение (путём указания причин наблюдаемых значений признаков).

В связи с тем, что приведённая модель онтологии учитывает большое число связей между процессами, происходящими в организме больного, можно ожидать, что алгоритм решения сформулированной выше задачи медицинской диагностики, анализирующий все эти связи, будет иметь высокую вычислительную сложность. Одним из путей повышения эффективности такого алгоритма является его распараллеливание и выполнение на многопроцессорной вычислительной системе. Наибольшего эффекта от распараллеливания можно достичь при решении некоторой частной, но тем не менее практически важной задачи, при решении которой признаки можно анализировать независимо друг от друга и, таким образом, параллельно на узлах многопроцессорного вычислительного комплекса. Такая задача получается при наложении на используемую модель онтологии нескольких ограничений: пациент может быть либо здоров, либо болен только одним заболеванием, а каждое заболевание имеет ровно один период развития.

В работе [16] приводится постановка задачи в терминах упрощённой онтологии и алгоритм её решения, а в работе [17] приводится распараллеленный алгоритм решения этой задачи, а также описана система, выполняющая оптимизирующее преобразование базы знаний о заболеваниях, в результате чего возможно сокращение количества гипотез о диагнозе. Результаты экспериментального исследования временной сложности оптимизированного алгоритма решения частной задачи медицинской диагностики, описанного в [17] приведены в работе [18]. Они показали, что:

а) использование оптимизирующего преобразования базы знаний о заболеваниях заметно ускоряет процесс постановки диагноза, в особенности – при наличии большого количества заболеваний в базе знаний;

б) наибольшая скорость постановки диагноза наблюдается при выполнении не менее одного и не более двух процессов на каждом узле кластера;

в) время работы алгоритма на всех использованных тестовых наборах данных при запуске оптимального числа процессов и использовании оптимизации базы знаний не превышает нескольких минут.

Описанная онтология и основанный на ней алгоритм медицинской диагностики могут быть использованы при создании системы, которая должна обеспечивать процесс согласованного решения комплекса задач по сбору, формализации, переводу в машиночитаемое представление, инженерии, хранению, управлению и обработке данных и знаний в области медицинской диагностики и является объединением всей этой информации в единый ресурс с возможностью удаленного доступа к нему многим пользователям.

Разрабатываемый ресурс назовем Банком знаний по медицинской диагностике. Для реализации указанных требований, архитектура этой системы должна соответствовать парадигме трёхзвенного программного обеспечения, т.е. банк знаний должен состоять из следующих трёх частей:

- информационного наполнения с некоторым стандартным интерфейсом доступа к хранимой информации и унифицированным форматом хранения этой информации;
- программного наполнения, ориентированного на интеллектуальную поддержку пользователей банка и включающего: средства по редактированию данных из информационного наполнения, средства по их обработке (оптимизация базы знаний о заболеваниях и медицинская диагностика пациента) и административную подсистему;
- интерфейсов к программным средствам из программного наполнения.

2. Информационное наполнение Банка знаний по медицинской диагностике

В информационном наполнении банка в первую очередь должна храниться модель онтологии предметной области медицинская диагностика [13,14,15]. В соответствии с ней в банк заносятся знания предметной области (описание заболеваний) и данные пациентов.

Модель онтологии, хранящаяся в информационном наполнении специализированного банка знаний, должна состоять из трёх частей:

- модель наблюдений,
- модель знаний о том, как заболевания влияют на значения признаков (модель знаний о заболеваниях),
- модель пациента (модель истории болезни пациента).

Модель онтологии базы наблюдений описывает структуру наблюдений и их значений. К наблюдениям относят наблюдаемые признаки, события и анатомо-физиологической особенности. На основе этой модели строится база наблюдений, куда входит описание названий событий, признаков и особенностей, а также перечисление всех их возможных значений.

Модель онтологии базы пациентов строится на основе модели онтологии базы наблюдений и задаёт структуру описания состояние пациента во времени. События могут происходить с пациентом в различные моменты времени и иметь различные значения. Признаки также в различные моменты времени могут иметь различные значения, а анатомо-физиологические особенности пациента имеют постоянные во времени значения. На основе данной модели пользователи, выполняющие медицинскую диагностику, формируют базу историй болезней пациентов, т.е. создают в информационном наполнении записи о пациентах: наблюдаемых у них в различные моменты времени значениях признаков, происходящих с ними событий и значения анатомо-физиологических особенностей.

В модели знаний о заболеваниях должны быть описаны в терминах модели наблюдений основные термины знаний (включая отношения между ними), в том числе заболевания (включая их причины) и нормальное состояние пациента. На основе схемы, задаваемой этой моделью, экспертами описываются

конкретные заболевания (значения признаков при заболеваниях задаются клиническими проявлениями и клиническими проявлениями, изменёнными воздействием событий), их причины (задаются этиологиями) и нормальное состояние пациента (описываются нормальные реакции и реакции на воздействие событий).

На рис. 1 представлена общая схема информационного наполнения банка знаний по медицинской диагностике. Стрелки указывают, как один компонент информационного наполнения используется при формировании другого.

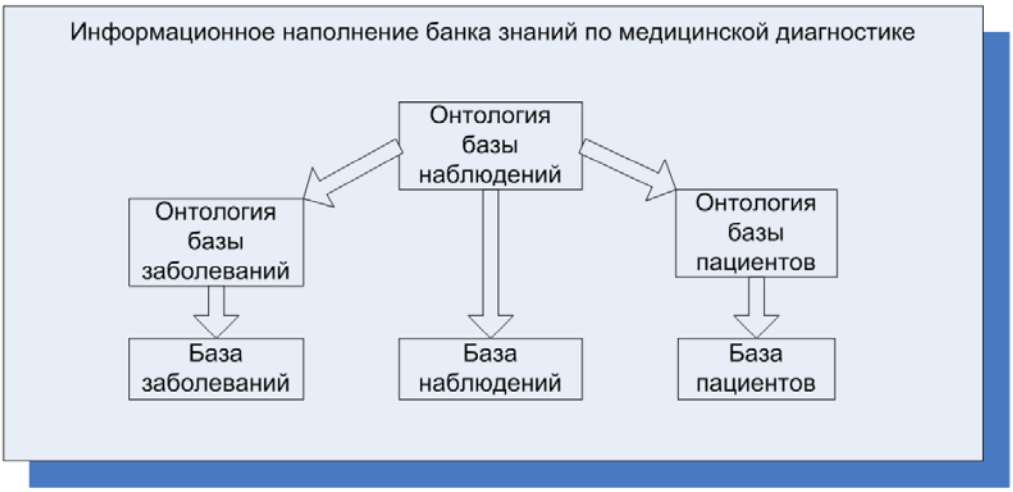


Рис. 1. Информационное наполнение банка знаний по медицинской диагностике.

3. Задачи, решаемые с помощью банка знаний пользователями различных типов, и соответствующее им программное наполнение

Как отмечалось в разделе 1, к программному наполнению разрабатываемого банка знаний относятся программные средства трёх типов:

- редакторы информационного наполнения,
- решатели задач,
- административная система.

В соответствии с приведёнными в разделе 2 составляющими информационного наполнения банка знаний по медицинской диагностике необходимо разработать следующие редакторы информационного наполнения:

- редактор онтологии наблюдений;
- редактор онтологии базы заболеваний (описания значений признаков при заболеваниях и у здорового пациента);
- редактор онтологии историй болезней пациентов;
- редактор базы наблюдений;
- редактор базы знаний о заболеваниях;
- редактор базы историй болезней пациентов.

При помощи редакторов онтологий на начальном этапе реализации банка знаний задаются модели соответствующих онтологий. Впоследствии данные средства не предполагается использовать, так как задание баз знаний необходимо вести в рамках жёстко закрепленных моделей онтологий.

В качестве алгоритма для решателя задачи медицинской диагностики используется упомянутый выше алгоритм [17], учитывающий все связи между знаниями и действительностью, заложенные в модели онтологии [14,15].

Программное средство, выполняющее оптимизирующее преобразование базы знаний о заболеваниях, должно работать по алгоритму, описанному в [17]. Оно анализирует информацию из базы заболеваний и определяет, при каких заболеваниях может наблюдаться каждое значение каждого признака. На основе этой информации алгоритм диагностики при формировании гипотез о заболевании формирует некоторое множество гипотез, возможно, меньшее, чем количество всех заболеваний в базе.

Всех пользователей банка знаний в области медицинской диагностики делятся на:

- обслуживающих пользователей (администратор и инженер-программист),
- носителей информации (экспертов-врачей),
- прикладных пользователей (врачей-диагностов и гостей).

Администратор следит за функционированием всего банка знаний при помощи административной подсистемы, выполняющей две функции: работа с пользовательскими записями и контроль над деятельностью пользователей.

Инженер-программист ответственен за функциональность всех компонент банка знаний. Он выполняет подготовку банка к функционированию и устраняет сбои в его работе, при возникновении таковых.

Врачи-эксперты занимаются редактированием баз наблюдений и знаний о заболеваниях, выполняя это в соответствии с моделью соответствующих онтологий. Такой пользователь для индивидуальной работы может создавать свою копию ИН или его части, где он может производить любые изменения, необходимые ему для работы. Если в процессе своей исследовательской или экспериментальной деятельности он получил существенные результаты, пополняющие базу данных, то он посылает запрос на пополнение информационного наполнения администратору, который принимает решение о возможности ее добавления в банк посредством административной системы.

Прикладные пользователи делятся на врачей, работающих с базой историй болезней и проводящих диагностику, и гостей, способных только знакомится с содержанием базы наблюдений и базы знаний о заболеваниях, находящихся в информационном наполнении банка.

4. Реализация Банка знаний по медицинской диагностике

Реализация требуемой системы может быть выполнена с использованием разработанного в ИАПУ ДВО РАН Многоцелевого Банка Знаний (МБкЗ) [19,20], предназначенного для поддержки жизненного цикла совместимых систем обработки информации.

В таком случае, программное наполнение Банка знаний по медицинской диагностике включает в себя:

- редактор ИРУО, являющийся частью МБкЗ, который используется для редактирования на языке ИРУО моделей онтологий и целевой информации, формируемой по этим моделям [21,22];
- административную систему, также входящую в состав МБкЗ, с помощью которой можно выполнять управление пользователями;
- решатель задачи медицинской диагностики;
- преобразователь базы знаний о заболеваниях.

Алгоритм диагностики реализован в виде параллельного приложения на языке C++, которое выполняется на многопроцессорном вычислительном комплексе (кластере) под управлением ОС Linux. Доступ к данному ресурсу предоставляется серверу банка знаний. На сервере банка знаний работает слабо-функциональная серверная часть решателя, реализованная на языке Java, которая ведёт общение с пользователем через интерфейсную часть и выполняет запуск диагностики на МВС с последующей трансляцией её результатов к клиенту-пользователю.

Поскольку алгоритм диагностики для быстрой постановки диагноза должен иметь быстрый доступ к знаниям и данным с как можно меньшим количеством промежуточных трансляционных узлов/прослоек, то

до начала диагностики необходимо отправить на кластер наполнение базы знаний о заболеваниях, для чего в подсистему оптимизации этой базы знаний встроен выполняющий это код. При начале диагностики, из информационного наполнения банка на кластер транслируется лишь история болезни пациента.

По окончании диагностики, все полученные результаты (опровергнутые диагнозы, подтверждённые диагнозы с их причинами и объяснениями значений наблюдаемых признаков) транслируются с кластера сервер-частью решателя в интерфейсную клиент-часть и отображаются на экране пользователя.

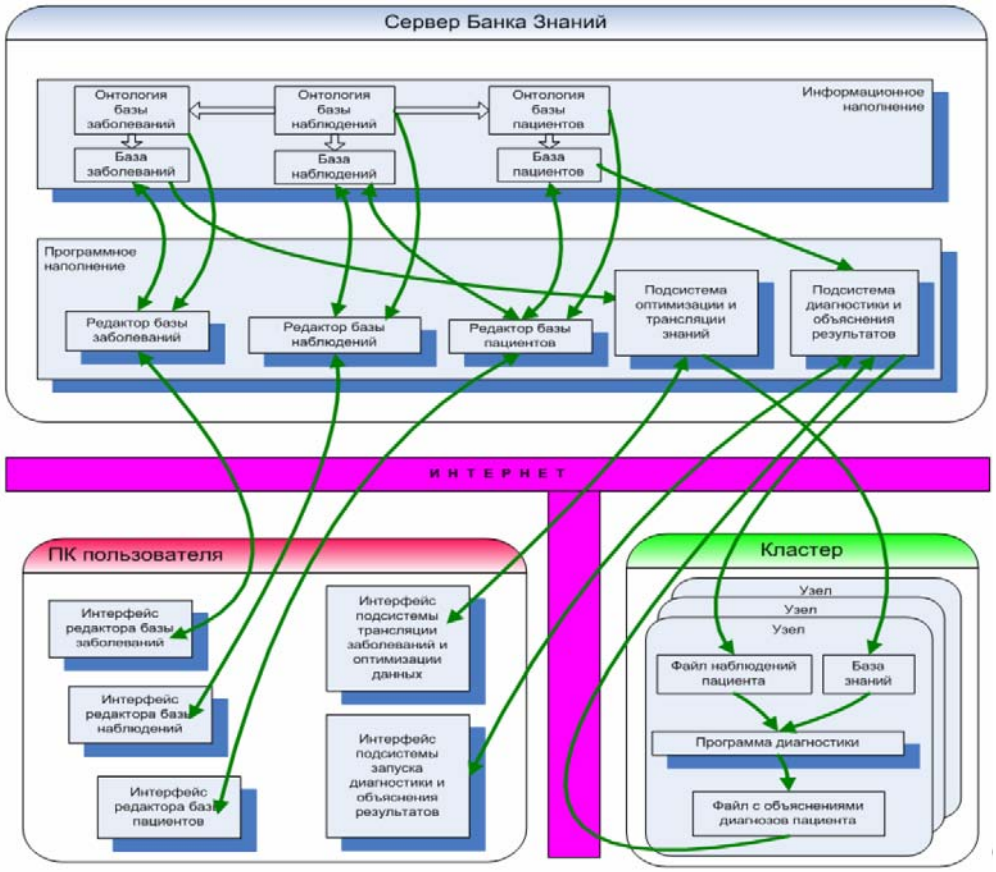


Рис.2. Архитектура банка знаний по медицинской диагностике.

На рис. 2 представлена общая архитектура банка знаний по медицинской диагностики, выполненная в рамках концепции СБк3. Ввиду сложности схемы на ней не отображается административная оболочка и её интерфейс, включая хранилище информации о пользователях банка, а также редакторы онтологий (в данном случае – Редактор ИРУО), так как они используются только на начальном этапе существования банка для задания моделей этих онтологий. Кроме того, следует понимать, что в качестве редактора баз знаний также используется Редактор ИРУО, но для каждой базы может быть разработан и включён в состав программного наполнения банка самостоятельный редактор, более удобный пользователям.

Заключение

В работе предложена концепция банка знаний по медицинской диагностике. Он содержит модель онтологии предметной области, состоящую из трёх частей (модель наблюдений, модель знаний о заболевании, модель истории болезни пациента), знания предметной области (база наблюдений и база знаний о заболеваниях) и данные действительности (база историй болезней пациентов). Кроме того, в банк входит средство редактирования моделей онтологий, баз знаний и данных, решатель задачи медицинской диагностики, программное средство, выполняющее оптимизацию базы знаний о

заболеваниях, и административная подсистема. Предложены пути реализации данного банка в рамках проекта Многоцелевого Банка Знаний, разработанного в ИАПУ ДВО РАН.

Список литературы

1. Генкин А.А. О последовательной стратегии Байеса и механизме принятия решений в интеллектуальной системе ОМИС. 1998. http://intels.spb.ru/english_med_informatiks-2.html
2. Detmer W.M., Shortliffe E.H. Using the Internet to Improve Knowledge Diffusion in Medicine. 1997. http://www-smi.stanford.edu/pubs/SMI_Reports/SMI-97-0658.pdf
3. Алексеев А.В. Применение методов информатики и компьютерной техники в дифференциальной диагностике аппендицита и аппендикулярной колики. http://arkadyal.chat.ru/programs.htm
4. Бурдаев В.П. Прикладные экспертные системы, основанные на оболочке "TECHEXP". 2002. http://uacm.kharkov.ua/eng/index.shtml?eexpert.htm
5. ООО "Админ". Экспертная система "Диагностика коматозных состояний". 1992. http://www.adminru.com/start_e.htm
6. Chabat F., Hansell D.M., Guang-Zhong Yang. Computerized Decision Support in Medical Imaging. 1997. http://www.doc.ic.ac.uk/~gzy/pub/chabat-decision-support.pdf
7. Гаськов А.П., Валивач М.Н. Экспертная система медицинской диагностики "Консилиум". 2000. http://www.eksi.kz/consilium/librar/esmd_short.htm
8. Lhotska L., Vlcek T. Efficiency enhancement of rule-based expert systems. Proceedings of the 15th IEEE symposium on computer-based medical systems (CBMS 2002). pp.53-58.
9. Matsumoto T., Ueda Y., Kawaji Sh. A software system for giving clues of medical diagnosis to clinician. Proceedings of the 15th IEEE symposium on computer-based medical systems (CBMS 2002). pp. 65-58.
10. Filho M.M., Palombo C.R., Sabbatini R.M. TMJ Plus: A Knowledge Base and Expert System for Diagnosis and Therapeutics of the Temporomandibular Joint Disorders. 1994. http://www.epub.org.br/ojdom/vol03n03.htm
11. Sawar S.J. Diagnostic Decision Support System of POEMS. 1999. http://www.cbl.leeds.ac.uk/sawar/projects/poems/poems-in-detail.html
12. Зильбер А.П., Шифман Е.М., Павлов А.Г., Белоусов С.Е. Интернет Проект "компьютерная диагностика преэклампсии". 1998. http://critical.onego.ru/critical/medlogic/
13. Chernyakhovskaya M.Yu., Kleschev A.S., Moskalenko F.M. A metaontology for medical diagnostics of acute diseases. Part 1. An informal description and definitions of basic terms. IJ ITA, 2007. (в печати)
14. Chernyakhovskaya M.Yu., Kleschev A.S., Moskalenko F.M. A metaontology for medical diagnostics of acute diseases. Part 2. A formal description of cause-and-effect relations. IJ ITA, 2007. (в печати)
15. Chernyakhovskaya M.Yu., Kleschev A.S., Moskalenko F.M. A metaontology for medical diagnostics of acute diseases. Part 3. A formal description of the causes of signs' values and of diseases. IJ ITA, 2007. (в печати)
16. Москаленко Ф.М. Задача медицинской диагностики и алгоритм ее решения, допускающий распараллеливание // Информатика и системы управления. – 2005. – № 2(10). – С.52-63.
17. Москаленко Ф.М. Параллельный оптимизированный алгоритм медицинской диагностики // Информатика и системы управления. – 2006. – № 1(11). – С.87-98.
18. Москаленко Ф.М. Экспериментальное исследование временной сложности параллельного алгоритма диагностики, основанного на реальной онтологии медицины // Информатика и системы управления. – 2006. – № 2(12). – С.42-53.
19. Орлов В.А., Клещев А.С. Компьютерные банки знаний. Многоцелевой банк знаний. // Информационные технологии – 2006. – №2. – С.2-8.
20. Орлов В.А., Клещев А.С. Компьютерные банки знаний. Требования к Многоцелевому банку знаний. // Информационные технологии – 2006. – №4. – С.21-28.
21. Орлов В.А., Клещев А.С. Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации. // Информационные технологии – 2006. – №5. – С.25-31.
22. Орлов В.А., Клещев А.С. Компьютерные банки знаний. Модель процесса редактирования информационного наполнения. // Информационные технологии – 2006. – №7. – С.11-16.

Информация об авторах

Москаленко Ф.М. - philipmm@yahoo.com

Клещев А.С. - kleschev@iacp.dvo.ru

Черняховская М.Ю. - chernyah@iacp.dvo.ru

Институт автоматики и процессов управления, Дальневосточное отделение Российской Академии Наук, Российская Федерация, г. Владивосток, ул.Радио, д.5.

INFORMATION-ANALYTICAL SYSTEM FOR DESIGN OF NEW INORGANIC COMPOUNDS

Nadezhda Kiselyova, Andrey Stolyarenko, Vladimir Ryazanov, Vadim Podbel'skii

Abstract*: The principles of design of information-analytical system (IAS) intended for design of new inorganic compounds are considered. IAS includes the integrated system of databases on properties of inorganic substances and materials, the system of the programs of pattern recognition, the knowledge base and managing program. IAS allows a prediction of inorganic compounds not yet synthesized and estimation of their some properties.*

Keywords*: information-analytical system, knowledge discovery in databases, design of new inorganic compounds, pattern recognition, computer learning, knowledge base, databases on properties of inorganic substances and materials.*

ACM Classification Keywords*: J.6 Computer-aided Design, J.2 Computer Applications&Chemistry, H.2.8 Scientific Databases, I.2.6 Analogies, I.2.6 Learning, I.2.4 Knowledge Representation Formalisms and Methods, C.2.5 Internet.*

Introduction

The problem of prediction of formation of new compounds and calculation of their properties is one of the most important tasks of inorganic chemistry. Any successful attempt of design of compounds not yet synthesized is of the large theoretical and practical importance. The problem of design of new inorganic compounds can be formulated as follows: it is necessary to find a combination of chemical elements and their ratio (that is, qualitative and quantitative composition) for making (under the given conditions) the predefined space molecular or crystal structure of compound allowing a realization of necessary functional properties. Only properties of chemical elements and data about other already investigated compounds should be used as initial information for calculations. Thus, the problem is concerned with a search for regularities between properties of chemical systems (for example, properties of compounds) and properties of elements, which form these systems.

The decision of a task of design of new inorganic compounds presents severe difficulties. The main difficulty is an extreme complexity of dependences relating property of inorganic compounds with properties of chemical elements. The traditional way of the decision of this task is associated with quantum-mechanical methods, which are based on the Schrodinger's equation. However in most cases the accurate solution (in analytical functions) of the latter for certain inorganic substances is fraught with great mathematical difficulties, which were been overcome only for the simplest systems. Therefore various approximated methods, as a rule, are used. These methods very much frequently do not give desirable results.

On the other hand, the chemistry had accumulated large information on properties of inorganic substances. There are periodic regularities between properties of compounds and properties of elements, which are included into their composition. This supposition is a consequence of the Periodic Law. Moreover, it is obviously, that already known compounds should be in accordance with these periodic regularities. The aims of our researches are development of methods and creation of computer system for search for these periodic regularities on the basis of analysis of information about already known substances accumulated in databases on properties of inorganic substances and materials. The found regularities are used for design of new inorganic compounds – analogues of already synthesized substances.

Selection of Methods of Search for Regularities in Information of Databases on Properties of Inorganic Substances and Materials

The methods of computer learning in pattern recognition are one of the most effective means of search for regularities in the large arrays of the chemical data [Kiselyova, 2005; Savitski and Gribulya, 1985]. In this case it is possible to connect some discrete parameters of inorganic compounds (for example, possibility of formation of compound or type of its crystal structure under normal conditions) with properties of elements, which are included into their composition, and also to get a threshold estimation of some numerical properties (for example, estimation of the melting point of compound at atmospheric pressure - above or below than certain threshold). It is important, that the fulfillment (though also not so strict) of basic hypothesis of methods of pattern recognition - hypothesis of compactness - is a consequence of the Periodic Law. Let an each compound corresponds to a point in multi-dimensional space of properties of elements. Owing to periodicity of properties of chemical elements points, which correspond to combinations of close on properties elements, combining into compounds, form compact clusters. Thus, the task of search for regularities connecting property of inorganic compounds with properties of chemical elements can be reduced to a problem of computer learning in pattern recognition. In this case the analysis of the information about already known compounds, which are represented as a set of values of properties of chemical elements, allows discovery of classifying regularities. The latter allow separation of known compounds into predetermined classes. It is possible to predict new compounds and estimate their unknown parameters by substitution of the property values of the appropriate chemical elements into the found regularities.

The principal problems at application of methods of pattern recognition to the decision of tasks of inorganic chemistry are following:

1. Small informativeness of attributes - properties of chemical elements.
2. The strong correlation of these attributes owing to their dependence on common parameter - atomic number of chemical elements (it follows from the Periodic Law).
3. Omissions in values of attributes.
4. In many cases - the large asymmetry of a size of classes of training set.
5. Sometimes feature description includes non-numerical attributes.
6. Possibility of experimental mistakes of classification in training sets.

In connection with the above-stated peculiarities of subject domain the search for methods and algorithms of pattern recognition allowing correct solution of these problems was one of the basic tasks of development of information-analytical system (IAS) for computer-aided design of inorganic compounds. It was established during testing various algorithms of computer learning for concrete tasks that it is impossible to specify beforehand, what algorithm is most effective at the decision of the certain chemical task of design of inorganic compounds. Quite often programs, which well have classified training set, obtained bad results at the prediction of unknown compounds. In this connection the most effective way of decision of tasks of predicting properties of new inorganic compounds is concerned with methods of recognition by collectives of algorithms [Zhuravlev et al., 2006]. At synthesis of the collective decision it is possible to compensate mistakes of separate algorithms by the correct predictions of other algorithms. Hence, the developed information-analytical system includes a set of the programs realized algorithms of various types, and also different strategies of collective decisions making.

Other way of increase of accuracy of predicting is a use of dependence of properties of chemical elements on atomic number. On the one hand, this fact complicates a task of search for separate properties that are the most important for classification of because of strong correlation of all used parameters of elements forming feature description. On the other hand, the classifying regularities including values of any subset of properties of chemical elements, which are used for the description of inorganic compounds, should in principle give identical results of classification. I.e. the results of the prediction with use of various subsets of properties of elements should, basically, coincide. This fact allows an additional possibility of collective decision making but already on the basis of collective of feature descriptions which was obtained as a result of division of initial set of properties of chemical elements on partially crossed subsets.

The problem of filling omissions also is partially solved with use of periodic dependences of parameters of elements. Replacement of the omission by average value of given parameter for two chemical elements that are nearest (within the range of group of Periodic System) to the element with omission is used.

After testing the programs the following software of pattern recognition were included into information-analytical system:

- a wide class of algorithms of system RECOGNITION developed by A.A.Dorodnicyn Computer Center of Russian Academy of Sciences (CCAS) [Zhuravlev et al., 2006]. This multifunctional system of pattern recognition includes the well-known methods of *k*–nearest neighbors, Fisher's linear discriminant, linear machine, multi-level perceptron (neural networks), support vector machine, genetic algorithm, and the special algorithms which were developed by CCAS: estimates calculation algorithms, LoReg (Logical Regularities), deadlock test algorithm, statistical weighted syndromes, etc.

- system of concept formation ConFor developed by V.M.Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine [Gladun, 1995, 2000, 2005]. The system is based on special data structure in a computer memory named as growing pyramidal networks.

It is important, that system RECOGNITION [Zhuravlev et al., 2006] is equipped with a set of algorithms of the decision of tasks of recognition by collectives of various algorithms. In this case task of recognition is decided in two stages. At first various algorithms, which are included into system, are applied independently. Further an optimum collective decision is made automatically with the help of special methods - "correctors". Some of methods of synthesis of the collective decisions - Bayesian corrector, convex stabilizer, some heuristic methods, etc. are used as correctors.

Databases and Knowledge Base of Information-Analytical Systems

The information basis of IAS (fig.1) is the integrated system of databases on properties of inorganic substances and materials [Dudarev et al., 2006; Kiselyova et al., 2005], which now includes:

- DBs containing the brief information on the most widespread properties of inorganic compounds and chemical elements:

1). DB on properties of inorganic compounds "Phases" [Kiselyova, 2005; Kiselyova et al., 2006] which now contains the information on properties more than 43, 000 ternary compounds (i.e. compounds formed by three chemical elements) and more than 15, 000 quaternary compounds, that was extracted from about 20, 000 publications.

2). DB on properties of chemical elements "Elements" which includes the data on more than 90 parameters.

- Specialized DBs which contain the detailed information on industrially vital substances and materials:

1). DB of phase diagrams of systems with intermediate semiconducting phases “Diagram” [Kiselyova, 2005; Khristoforov et al., 2001], that contains information on the most important pressure-temperature-concentration phase diagrams of semiconducting systems evaluated by qualified experts and also on the physical-chemical properties of the intermediate phases. Now DB contains the detailed information on several tens binary and ternary systems extracted from 2000 publications.

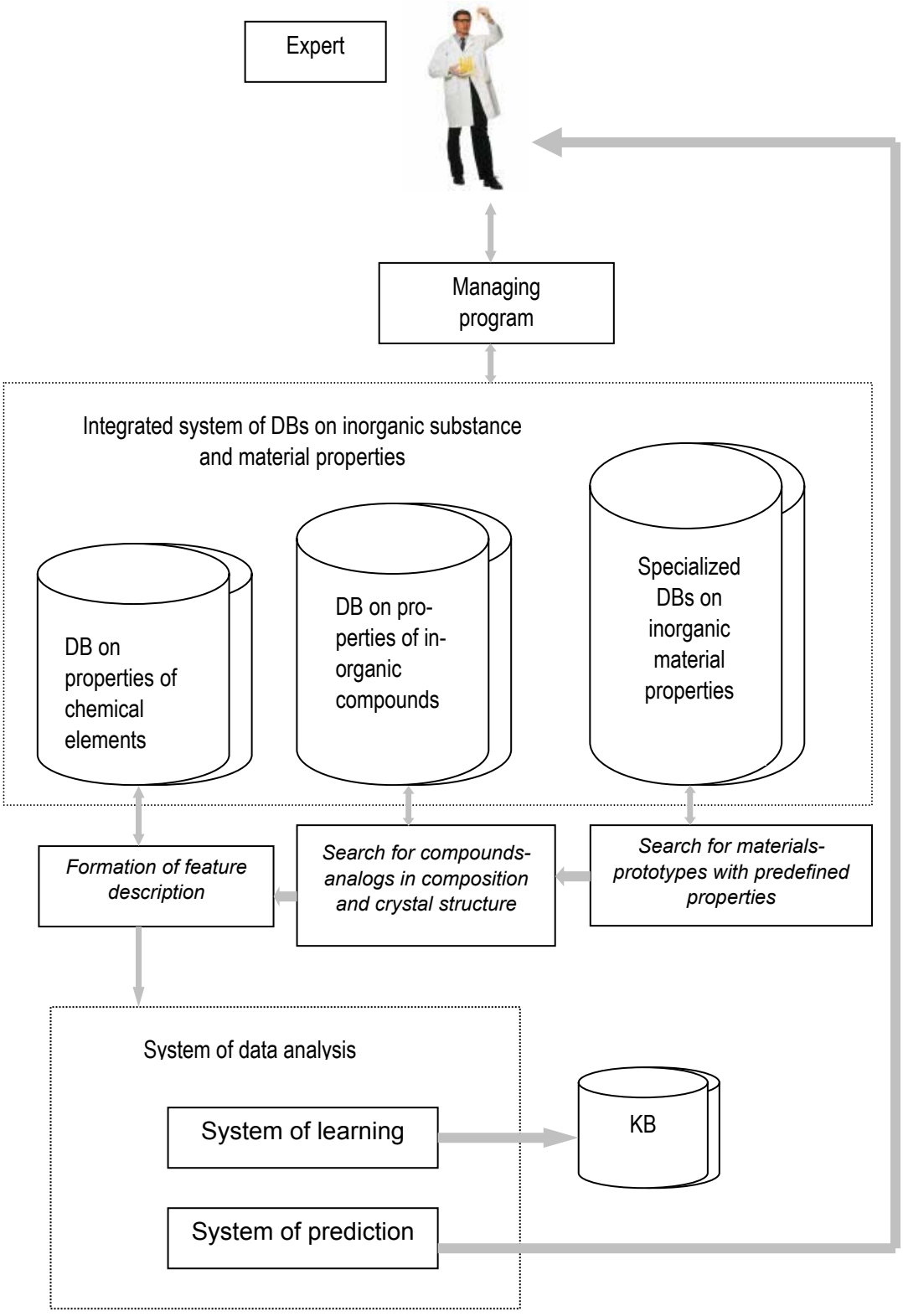


Fig.1. Principal schema of IAS

2). DB on substances with significant acousto-optical, electro-optical and nonlinear-optical properties “Crystal” [Kiselyova, 2005; Kiselyova et al., 2004] which now includes the information on parameters more than 100 materials.

3). DB on width of the forbidden zone of inorganic substances “Bandgap” [Dudarev et al., 2006] which now contains the data on more than 2, 000 substances.

Cumulative volume of DBs is ~7 GB. All these databases are accessible from Internet (www.imet-db.ru).

The of knowledge base (KB) of information-analytical system includes the relational tables containing regularities found during computer learning with the indication about common chemical composition of compounds, set of attributes included into regularity, parameter to be predicted, used algorithm, and also service information (date of updating, surname of the expert who carried out computer learning, etc.). Both the integrated system of DBs on properties of inorganic substances and materials and knowledge base are realized with use of DBMS MS SQL Server. DBs, which are incorporated into the integrated information system, use various DBMS [Dudarev et al., 2006].

The information-analytical system for design of inorganic compounds is intended for users of two levels. Firstly it is a reference system for ordinary specialist. Secondly IAS is a tool for expert estimating the chemical information for computer learning and carrying out a search for regularities in data (fig.1). In last case owing to use of knowledge and experience of the highly skilled experts the mentioned above problems of selection of the most important attributes for the description of compounds and filtration of mistakes of classification of objects of training set are partially decided.

Program Realization of Information-Analytical System

Feature of program realization of IAS is the design of client module completely on the basis of the Web-interface. The users work with IAS using only Web-browser. Thus, the users do not need to install of any additional programs. It also facilitates an expansion of system by new methods and functionalities: the changes are done only in server where the system is located. The systems of learning and prediction are realized using C++ and are broken up into modules, which realize the rigidly fixed interface. The basic module is the managing program Engine.dll that is responsible for management of all processes in system, and for communication between modules. System also includes the uniform modules of data analysis realizing various mathematical methods of pattern recognition. The important feature of organization of system is a wide use of dynamic linking. Each method is realized as separate dynamic library and can be modernized without changing other parts of the program. The interaction between subsystems of leaning and prediction and subsystem of the graphic Web-interface is executed as connection of program (in C#-code) of appropriate methods from dynamic library Engine.dll. The knowledge base is realized using SQL-server and Web-server. Web-server is intended for storing all necessary files using special format for subsequent their application to recognizing. SQL-server stores complete information on the obtained regularities.

Conclusion

The information-analytical system, created by us, allows solution of two important tasks of inorganic chemistry. First, it allows the partially automation of analysis of the huge experimental information, accumulated by chemistry, for search for regularities in the data and subsequent design of new compounds with predefined properties. Secondly, it expands opportunities of traditional DBs on properties of substances and materials, giving the user not only information on the already investigated substances, but also predictions of some substances not yet synthesized and estimation of their properties. Essential advantage of developed IAS is Internet-access. In

this case user receives operative access to "alive" data and regularities. With the help of IAS it was possible to predict some new inorganic compounds and to estimate their some properties.

Acknowledgements

Financial support from RFBR (grants №06-07-89120 and 05-03-39009) is gratefully acknowledged. We should like to thank our colleagues Prof. V.P.Gladun, Dr. V.Yu.Velichko of the Institute of Cybernetics of the National Academy of Sciences of Ukraine, Prof. V.S.Zemskov, Dr. V.A.Dudarev, Dr. I.V.Prokoshev, V.V.Khorbenko of the Institute of Metallurgy and Materials Science of Russian Academy of Sciences, and Dr. O.V.Sen'ko of the Computer Center of the Russian Academy of Sciences for their help and support.

Bibliography

[Dudarev et al., 2006] V.A.Dudarev, N.N.Kiselyova, V.S.Zemskov. Integrated system of databases on properties of materials for electronics. Perspektivnye Materialy, 2006, N.5 (Russ.).

[Gladun, 1995] V.P.Gladun. Processes of Formation of New Knowledge. SD "Pedagog 6", Sofia, 1995 (Russ.).

[Gladun, 2000] V.P.Gladun. Partnership with Computer. Port-Royal. Kiev, 2000 (Russ.).

[Gladun, 2004] V.P.Gladun. Growing pyramidal networks. Novosti Iskusstvennogo Intellekta, 2004, №1 (Russ.).

[Khristoforov et al., 2001] Yu.I.Khristoforov, V.V.Khorbenko, N.N.Kiselyova, et al. Internet-accessible database on phase diagrams of semiconductor systems. Izvestiya VUZov. Materialy Elektron.Tekhniki, 2001, №4 (Russ.).

[Kiselyova, 2005] N.N.Kiselyova. Computer Design of Inorganic Compounds. Application of Databases and Artificial Intelligence. Nauka, Moscow, 2005 (Russ.).

[Kiselyova et al., 2005] N.N.Kiselyova, V.A.Dudarev, I.V.Prokoshev, et al. The distributed system of databases on properties of inorganic substances and materials. Int.J."Information Theories & Applications", 2005, v.12.

[Kiselyova et al., 2006] N.Kiselyova, D.Murat, A.Stolyarenko, et al. Database on ternary inorganic compound properties "Phases" in Internet. Informazionnye Resursy Rossii, 2006, N.4 (Russ.).

[Kiselyova et al., 2004] N.N.Kiselyova, I.V.Prokoshev, V.A.Dudarev, et al. Internet-accessible electronic materials database system. Inorganic Materials, 2004, v.42, №3.

[Savitski and Gribulya, 1985] E.M.Savitski and V.B.Gribulya. Application of Computer Techniques in the Prediction of Inorganic Compounds. Oxonian Press Pvt.Ltd., New Delhi-Calcutta, 1985.

[Zhuravlev et al., 2006] Yu.I.Zhuravlev, V.V.Ryazanov, O.V.Senko. RECOGNITION. Mathematical methods. Software System. Practical Solutions. Phasis, Moscow, 2006 (Russ.).

Authors' Information

Nadezhda Kiselyova *– A.A.Baikov Institute of Metallurgy and Materials Science of Russian Academy of Sciences, P.O.Box: 119991 GSP-1, 49, Leninskii Prospect, Moscow, Russia, e-mail: kis@ultra.imet.ac.ru*

Andrey Stolyarenko *–- A.A.Baikov Institute of Metallurgy and Materials Science of Russian Academy of Sciences, P.O.Box: 119991 GSP-1, 49, Leninskii Prospect, Moscow, Russia, e-mail: stol-drew@yandex.ru*

Vladimir Ryazanov *– A.A.Dorodnicyn Computer Center of Russian Academy of Sciences, e-mail: riazanov@ccas.ru*

Vadim Podbel'skii *– Moscow Institute of Electronics and Mathematics (Technical University), P.O.Box: 109028, B.Trehsvjatitelsky per. 3/12, Moscow, Russia, e-mail: vvp@mitme.ru, vpodbelskiy@hse.ru*

A MODEL OF RULE-BASED LOGICAL INFERENCE

Xenia Naidenova

Abstract: *We proposed a unified model for combining inductive reasoning with deductive reasoning in the framework of inferring and using implicative logical rules. The key concept of our approach is the concept of a good diagnostic test. We define a good diagnostic test as the best approximation of a given classification on a given set of examples. The analysis of the good tests construction allows demonstrating that this inference engages both inductive and deductive reasoning rules.*

Key Words *learning logical rules from examples, machine learning, rule-based inference, good diagnostic test*

Introduction

Here we describe a model of common reasoning that has been acquired from our numerous investigations on the human reasoning modes used by experts for solving diagnostic problems in diverse areas such as pattern recognition of natural objects (rocks, ore deposits, types of trees, types of clouds e.t.c.), analysis of multi-spectral information, image processing, interpretation of psychological testing data, medicine diagnosis and so on. The principal aspects of this model coincide with the rule-based inference mechanism that is embodied in the KADS system (Ericson, et al., 1992), (Gappa and Poeck, 1992). More details related to our model of reasoning and its implementation can be found in (Naidenova and Syrbu, 1984), (Naidenova and Polegaeva, 1985a), (Naidenova and Polegaeva, 1985b), and (Naidenova, 2006).

We need the following three types of rules in order to realize logical inference:

INSTANCES or relationships between objects or facts really observed. Instance can be considered as a logical rule with the least degree of generalization. On the one hand, instances serve as a source of an expert's knowledge. On the other hand, instances can serve as a source of a training set of positive and negative examples for inductive inference of generalized rules.

RULES OF THE FIRST TYPE. These rules describe regular relationships between objects and their properties and between properties of different objects. The rules of the first type can be given explicitly by an expert or derived automatically from examples with the help of some learning process. These rules are represented in the form "if-then" assertions. They accumulate generalized knowledge in a problem domain.

RULES OF THE SECOND TYPE or inference rules with the help of which rules of the first type are used, updated and inferred from data (instances). The rules of the second type are reasoning rules.

Using the rules of the first type is artificially separated from the learning process. But it is clear that there is no reason to separate the process of learning rules from the process of using these rules for class identification or pattern recognition problems: both processes are interdependent and interconnected. Anyone of these processes can require executing the other. Anyone of these processes can be built into the other.

Any model of reasoning must also include STRATEGIES or the sequences of applying rules of all types in reasoning. The application of rules is conditioned by different situations occurring in the reasoning process and it is necessary to identify these situations. Strategies have a certain freedom as so it is possible to apply different rules in one and the same situation and the same rule in different situations. The choice of a strategy determines the speed, completeness, deepness and quality of reasoning.

Rules Acquired from Experts or Rules of the First Type

An expert's rules are logical assertions that describe the knowledge of specialists about a problem domain. Our experience in knowledge elicitation from experts allows us to analyze the typical forms of assertions used by experts. As an example, we give the rules of an expert's interpretation of data obtained with the use of

Pathological Character Accentuation Inventory for Adolescents. This psycho-diagnostic method was elaborated by Lichko (1983) and is a classical example of an expert system.

Some examples of the expert's rules are:

"If (D - F) $\geq$ 4, then DISSIMULATION decreases the possibility to reveal any character accentuation and completely excludes the CYCLOID and CONFORM types of character".

"If the index E $>$ 4, then the CYCLOID and PSYCHASTENOID types are impossible".

"If the type of character is HYPERTHYMIA, then ACCENTUATION with psychopathies is observed in 75%, with transit disturbances – in 5%, and with stable adaptation – in 5% of all cases".

"If the index A $>$ 6 and the index S $>$ 7 and the index Con = 0 and the index D $>$ 6, then the LABILE type is observed".

"If the index E $\geq$ 6, then the SCHISOID and HYSTEROID types are observed frequently".

"If after the application of rules with the numbers *x*, *y*, *z* the values of at least two indices are greater than or equal to the minimal diagnostic threshold, then the mixed types are possible with the following consistent combinations of characters: Hyp - C, Hyp - N, Hyp - Hyst, C - L, L - A, L - S, and L - Hyst".

We used the following abbreviations: Hyp - hyperthymia, C - cycloid, L - labile, A – asthenia, N – neurotic, S - schizoid, Con - conformable, Hyst - hysteroid, Sens - sensitive, D - dissimulation, F - frankness, E - emancipation, and P - psychasthenia.

It is clear that an expert's assertions can be represented with the use of only **one class of logical rules**, namely, the rules based on implicative dependencies between names.

Implication: *a*, *b*, *c* $\rightarrow$ *d*. This rule means that if the values standing in the left side of the rule are simultaneously true, then the value in the right side of the rule is always true.

An implication *x* $\rightarrow$ *d* is satisfied if and only if the set of situations in which *x* appears is included in the set of situations in which d appears.

Interdiction or forbidden rule (a special case of implication) *a*, *b*, *c* $\rightarrow$ false (never). This rule interdicts a combination of values enumerated in the left side of the rule. The rule of interdiction can be transformed into several implications such as *a*, *b* $\rightarrow$ not *c*; *a*, *c* $\rightarrow$ not *b*; and *b*, *c* $\rightarrow$ not *a*.

Compatibility: *a*, *b*, *c* $\rightarrow$ *rarely*; *a*, *b*, *c* $\rightarrow$ *frequently*. This rule says that the values enumerated in the left side of the rule can simultaneously occur rarely (*frequently*). The rule of compatibility presents the most frequently observed combination of values that is different from a law or regularity with only one or two exceptions.

Compatibility is equivalent to a collection of assertions as follows:

a, *b*, *c* $\rightarrow$ *rarely* *a*, *b*, *c* $\rightarrow$ *frequently*,

a, *b* $\rightarrow$ *c rarely* *a*, *b* $\rightarrow$ *c frequently*,

a, *c* $\rightarrow$ *b rarely* *a*, *c* $\rightarrow$ *b frequently*,

b, *c* $\rightarrow$ *a rarely* *b*, *c* $\rightarrow$ *a frequently*.

Diagnostic rule: *x*, *d* $\rightarrow$ *a*; *x*, *b* $\rightarrow$ not *a*; *d*, *b* $\rightarrow$ *false*. For example, *d* and *b* can be two values of the same attribute. This rule works when the truth of '*x*' has been proven and it is necessary to determine whether '*a*' is *true* or not. If '*x* & *d*' is true, then '*a*' is *true*, but if '*x* & *b*' is true, then '*a*' is *false*.

Rule of alternatives: *a* or *b* $\rightarrow$ *true* (*always*); *a*, *b* $\rightarrow$ *false*. This rule says that '*a*' and '*b*' cannot be simultaneously true, either '*a*' or '*b*' can be true but not both.

Structure of the Knowledge Base

We describe a very simple structure of a knowledge base that is sufficient for our illustrative goal. The knowledge base (KB) consists of two parts: the Attribute Base (*AtB*), containing the relations between problem domain concepts, and the Assertion Base (*AsB*), containing the expert's assertions formulated in terms of the concepts.

The domain concepts are represented by the use of names. With respect to its role in the KB, a name can be one of two kinds: name of attribute and name of attribute value. However, with respect to its role in the problem domain, a name can be the name of an object, the name of a class of objects and the name of a classification or collection of classes. A class of objects can contain only one object hence the name of an object is a particular case of the name of a class. In the KB, names of objects and of classes of objects become names of attribute values, and names of classifications become names of attributes.

For example, let objects be a collection of trees such as asp, oak, fir-tree, cedar, pine-tree, and birch. Each name calls the class or the kind of trees (in a particular case, only one tree). Any set of trees can be partitioned into the separate groups depending on their properties. '*Kind of trees*' will be the name of a classification, in which '*asp*', '*oak*', '*fir-tree*', '*cedar*', '*pine-tree*', and '*birch*' are the names of classes. Then, in the KB, '*kind of trees*' will be used as the name of an attribute the values of which are '*asp*', '*oak*', '*fir-tree*', '*cedar*', '*pine-tree*', and '*birch*'. The link between the name of an attribute and the names of its values is implicative. It can be expressed by the following way:

(<name of value$_1$>, <name of value$_2$>, ... , <name of value $_k$ >) $\rightarrow$ <name of attribute>,

where the sign "$\rightarrow$" denotes the relation "is a".

In our example (*asp*, *oak*, *fir-tree*, *cedar*, *pine-tree*, *birch*) $\rightarrow$ *kind of trees*, and, for each value of '*kind of trees*', the assertion of the following type can be created: "*asp* is a *kind of trees*".

The set of all attributes' names and the set of all values' names must not intersect. This means that the name of a classification cannot simultaneously be the name of a class. However, this is not the case in natural languages: the name of a class can be used for some classification and vice versa. For example, one can say that '*pine-tree*', '*fir-tree*', '*cedar*' are '*conifers*'. But one may also say that '*conifers*', '*leaf-bearing*' are '*kinds of trees*'. Here the word '*conifers*' serves both as the name of a classification and as the name of a class. In this setting, class is a particular case of classification like object is a particular case of class.

By using names in the way we do in real life we permit the introduction of auxiliary names for the subsets of the set of an attribute's values. Let *A* be an attribute. The name of a subset of values of *A* will be used as the name of a new attribute which, in its turn, will serve as the name of a value with respect to *A*.

The *AsB* (Assertion Base) contains the expert's assertions. Each assertion links a collection of values of different attributes with a certain value of a special attribute (SA) that evaluates how often this collection of values appears in practice. The values of a special attribute are: *always*, *never*, *rarely*, and *frequently*. Assertions have the following form:

(<name of value>, <name of value>, … , <value of SA>) = *true*.

For simplicity, we omit the word '*true*', because it appears in any assertion. For example, the assertion "*pine-tree* and *cedar* can be found *frequently* in the meadow type of forest" will be expressed in the following way: (*meadow*, *pine-tree*, *cedar*, *frequently*). We also omit the sign of conjunction between values of different attributes and the sign of disjunction (separating disjunction) between values of the same attribute. For example, the assertion in the form (*meadow*, *pine-tree*, *cedar*, *frequently*) is equivalent to the following expression of formal logic: P((type of forest = *meadow*) & ((kind of trees = *pine-tree*) V (kind of trees = *cedar*)) & (SA = *frequently*)) = *true*.

Only one kind of requests to the KB is used: SEARCHING VALUE OF <name of attribute> [,<name of attribute>,...] IF (<name of value>, <name of value>, ...), where "name of value" is the known value of an attribute, "name of attribute" means that the value of this attribute is unknown. For example, the request "to find the type of forest for a region with plateau, without watercourse, with the prevalence of pine-tree" will be represented as follows: SEARCHING VALUE OF the type of forest IF (*plateau*, *without watercourse*, *pine-tree*).

Deductive Reasoning Rules of the Second Type

The following rules of the second type lie in the basis of the reasoning process for solving diagnostic or pattern recognition tasks. Let *x* be a collection of true values observed simultaneously.

Using implication. Let *r* be an implication, left(*r*) be the left part of *r* and right(*r*) be the right part of *r*. If left(*r*) $\subseteq$ *x*, then *x* can be extended by right(*r*): $x \leftarrow x \cup$ right(*r*).

For example, *x* = '*a*, *b*, *c*, *d* ', *r* = '*a*, *d* $\rightarrow$ k ', $x \leftarrow x \cup k$.

Using implication is based on **modus ponens**: if *A*, then *B*; *A*; hence *B*.

Using interdiction. Let *r* be an implication *y* $\rightarrow$ not *k*. If left(*r*) $\subseteq$ *x*, then k is a forbidden value for all the extensions of *x*.

Using interdiction is based on **modus ponendo tollens**:

either *A* or *B* (*A*, *B* – alternatives); *A*; hence not *B*;

either *A* or *B*; *B*; hence not *A*.

Using compatibility. Let *r* = '*a*, *b*, *c* $\rightarrow$ *k*, *rarely* (*frequently*)'.

If left(*r*) $\subseteq$ *x*, then *k* can be used for an extension of *x* with the value of SA equal to '*rarely*' ('*frequently*'). The application of several rules of compatibility leads to the appearance of several values 'rarely' and/or '*frequently*' in the extension of *x*. Computing the value of SA for the extension of *x* requires special consideration. In any case, the appearance of at least one value '*rarely*' means that the total result of the extension will have the value of SA equal to '*rarely*'. Two values equal to '*frequently*' lead to the result 'less *frequently*', three values equal to '*frequently*' lead to the result 'less less *frequently*' and hence the values '*rarely*' and '*frequently*' must have the ordering scale of measuring.

Using compatibility is based on **modus ponens**.

Using diagnostic rules. Let *r* be a diagnostic rule such as '*x*, *d* $\rightarrow$ *a*; *x*, *b* $\rightarrow$ not *a*', where '*x*' is true, and 'a', 'not a' are hypotheses or possible values of some attribute. There are several ways for refuting one of the hypotheses: to infer either *d* or *b* with the use of knowledge base (*AtB*, *AsB*); to involve new instances from the database and/or new assertions from the knowledge base for inferring new diagnostic rules distinguishing the hypotheses '*a*' and 'not *a*'; or, eventually, ask an expert which of the values *d* or *b* is true.

Our experience shows that generally the experts have in their disposal many diagnostic rules corresponding to the most difficult diagnostic situations in their problem domain.

Using a diagnostic rule is based on **modus ponens** and **modus ponendo tollens**.

Using rule of alternatives. Let '*a*', '*b*' be two alternative hypotheses about the values of some attribute. If one of these hypotheses is inferred with the help of reasoning operations, then the other one is rejected.

Using a rule of alternatives is based on **modus tollendo ponens**: either *A* or *B* (*A*, *B* – alternatives); not *A*; hence *B*; either *A* or *B*; not *B*; hence *A*.

The operations enumerated above can be named as "forward reasoning" rules. Experts also use implicative assertions in a different way. This way can be named as "backward reasoning".

Generating hypothesis. Let *r* be an implication *y* $\rightarrow$ *k*. Then the following hypothesis is generated "if *k* is true, then it is possible that *y* is true".

Using modus tollence. Let **r** be an implication *y* $\rightarrow$ *k*. If 'not *k* ' is inferred, then 'not *y*' is also inferred.

Natural diagnostic reasoning is not any method of proving the truth. It has another goal: to infer all possible hypotheses about the value of some target attribute. These hypotheses must not contradict with the expert's knowledge and the situation under consideration. The process of inferring hypotheses is reduced to extending maximally a collection *x* of attribute values such that none of the forbidden pairs of values would belong to the extension of *x*.

Inductive Reasoning Rules of the Second Type

Inductive steps of reasoning consist of using already known facts and statements, observations and experience for inferring new logical rules of the first type or correcting those that turn out to be false.

For this goal, inductive rules of reasoning are applied. The main forms of induction are the canons of induction that have been formulated by English logician Mill (1900). These canons are known as the five induction methods of reasoning: method of only similarity, method of only distinction, joint method of similarity-distinction, method of concomitant changes, and method of residuum.

The method of only similarity: This rule means that if the previous events (values) *A*, *B*, *C* lead to the events (values) *a*, *b*, *c* and the events (values) *A*, *D*, *E* lead to the events (values) *a*, *d*, *e*, then *A* is a reason of *a*.

The method of only distinction: This rule means that if the previous events (values) *A*, *B*, *C* give rise to the events (values) *a*, *b*, *c* and the events (values) *B*, *C* lead to the events (values) *b*, *c*, then *A* is a reason of *a*.

The joint method of similarity-distinction: This method consists of applying two previous methods simultaneously.

The method of concomitant changes: This rule means that if the change of a previous event (value) *A* is accompanied by the change of an event (value) *a*, and all the other previous events (values) do not change, then *A* is a reason of *a*.

The method of residuum: Let *abcd* be a complex phenomenon, *A* be the reason of *a*, *B* be the reason of *b*, and *C* be the reason of *c*. Then it is possible to suppose that there is an event *D* which is a reason of *d*.

An Example of the Reasoning Process

Let the content of the Knowledge Base be the following collection of assertions:

AtB:

1. (*meadow*, *bilberry wood*, *red bilberry wood* ....) → *types of woodland*;
2. (*pine-tree*, *spruce*, *cypress*, *cedars*, *birch*, *larch*, *asp*, *fir-tree*) → *dominating kinds of trees*;
3. (*plateau*, *without plateau*) → *presence of plateau*;
4. (*top of slope*, *middle part of slope*, ….) → *parts of slope*;
5. (*peak of hill*, *foot of hill*) → *parts of hill*;
6. (*height on plateau*, *without height on plateau*) → *presence of a height on plateau*;
7. (*head of watercourse*, *low part of watercourse*, ….) → *parts of water course*;
8. (*steepness* ≥ 4°, *steepness* ≤ 3°, *steepness* < 3°, …) → *features of slope*;
9. (*north*, *south*, *west*, *east*) → *the four cardinal points*;
10. (*watercourse*, *without watercourse*) → *presence of a watercourse*.

AsB:

11. (*meadow*, *pine-tree*, *larch*, *frequently*);
12. (*meadow*, *pine-tree*, *steepness* ≤ 4°, *never*);
13. (*meadow*, *larch*, *steepness* ≥ 4°, *never*);
14. (*meadow*, *north*, *west*, *south*, *frequently*);
15. (*meadow*, *east*, *rarely*);
16. (*meadow*, *fir-tree*, *birch*, *asp*, *rarely*);
17. (*meadow*, *plateau*, *middle part of slope*, *frequently*);
18. (*meadow*, *peak of hill*, *watercourse heads*, *rarely*);
19. (*plateau*, *steepness* ≤ 3°, *always*);
20. (*plateau*, *watercourse*, *rarely*);
21. (*red bilberry wood*, *pine-tree*, *frequently*);
22. (*red bilberry wood*, *larch*, *rarely*);
23. (*red bilberry wood*, *peak of hill*, *frequently*);
24. (*red bilberry wood*, *height on plateau*, *rarely*);
25. (*meadow*, *steepness* < 3°, *frequently*).

Let *x* be a request to the KB equal to:

SEARCHING VALUE OF type of woodland IF (*plateau*, *without watercourse*, *pine-tree*).

The process of reasoning evolves according to the following sequence of steps:

Step 1. Take out all the assertions *t* in *AsB* containing at least one value from the request, i.e. $t \in AsB$, $t \cap x \neq \varnothing$, where *x* is the request. These are assertions 11, 12, 17, 19, 20, 21, and 24.

Step 2. Delete (from the set of selected assertions) all the assertions that contradict the request. Assertion 20 contradicts the request because it contains the value of attribute '*presence of water course*' which is different from the value of this attribute in the request. The remaining assertions are 11, 12, 17, 19, 21, and 24. This step uses **the rule of alternatives**.

Step 3. Take out the values of attribute '*type of woodland* ' appearing in assertions 11, 12, 17, 19, 21, and 24. We have **two hypotheses**: '*meadow* ' and '*red bilberry* '. This step uses **implications** for generating **hypotheses**. It is a step of "forward reasoning". As a result, we have two extensions of the request:

SEARCHING VALUE OF the type of woodland IF (*plateau*, *without watercourse*, *pine-tree*, *meadow*?).

SEARCHING VALUE OF the type of woodland IF (*plateau*, *without watercourse*, *pine-tree*, *red bilberry*?).

The sign ' ? 'means that the values of type of woodland are hypotheses.

Step 4. An attempt is made to refute one of the hypotheses (the application of a **diagnostic rule**). For this goal, it is necessary to find an assertion that has the value of SA equal to '*never* ' and contains one of the hypotheses, some subset of values from the request and does not contain any other value. There is only one assertion with the value of SA equal to '*never* '. This is assertion 12: (*meadow*, *pine-tree*, *steepness* $\leq 4°$, *never*). However, we cannot use this assertion because it contains the value '*steepness* $\leq 4°$' which is not in the request.

Step 5. An attempt is made to find a value of some attribute that is not in the request (in order to extend the request). For this goal, it is necessary to find an assertion with the value of SA equal to '*always*' that contains a subset of values from the request and one and only one value of some new attribute the values of which are not in the request. Only one assertion satisfies this condition. This is assertion 19: (*plateau*, *steepness* $\leq 3°$, *always*).

Step 6. Forming the extended requests:

SEARCHING VALUE OF the type of woodland IF (*plateau*, *without watercourse*, *pine-tree*, *steepness* $\leq 3°$, *meadow*?).

SEARCHING VALUE OF the type of woodland IF (*plateau*, *without watercourse*, *pine-tree*, *steepness* $\leq 3°$, *red bilberry*?).

It is easy to see that Step 5 and Step 6 involve the rule of **using implication** in order to extend the requests.

Steps 1, 2, and 3 are repeated. Assertion 25 is involved in the reasoning.

Step 4 is repeated. Now assertion 12 can be used because the value '*steepness* $\leq 4°$ is in accordance with the values of '*feature of slope*' in the request. We conclude now that the type of woodland cannot be '*meadow*'. The non-refuted hypothesis is "the type of woodland = *red bilberry*". This step uses the interdiction rule in order to delete one of the hypotheses.

The process of pattern recognition can require **inferring new rules of the first type** from data, when it is impossible to distinguish inferred hypotheses. In general, there exist two main cases to learn rules of the first type from examples in the process of pattern recognition: i) the result of reasoning contains several hypotheses and it is impossible to choose one and only one of them (uncertainty), and ii) there does not exist any hypothesis.

An approach to Inferring Rules of the First type

Our approach to learning implicative rules from data is based on the concept of a good diagnostic (classification) test. A good classification test can be understood as an approximation of a given classification on a given set of examples (Naidenova and Polegaeva, 1986; Naidenova, 1996).

A good diagnostic test is defined as follows. Let R be a set of examples and $S = \{1, 2, \ldots i, \ldots, n\}$ be the set of indices of examples, where n is the number of example of R. Let $R(+)$ and $S(+)$ be the set of positive examples and the set of indices of positive examples, respectively. By $R(-) = R/R(+)$ denote the set of negative examples. Let U be the set of attributes and T be the set of attributes values (values, for short) each of which appears at least in one of the examples of R.

Denote by $s(A)$, $A \in T$ the subset $\{i \in S: A \text{ appears in } t_i, t_i \in R\}$, where $S = \{1, 2, .., n\}$.

Following (Cosmadakis, et al., 1986), we call $s(A)$ the interpretation of $A \in T$ in R. The definition of $s(A)$ can be extended to the definition of $s(t)$ for any collection t, $t \subseteq T$ of values as follows:

if $t = A_1 A_2 \ldots A_m$, then $s(t) = s(A_1) \cap s(A_2) \cap \ldots \cap s(A_m)$.

Definition 1. A collection $t \subseteq T$ ($s(t) \neq \varnothing$) of values is a diagnostic test for the set $R(+)$ of examples if and only if the following condition is satisfied: $t \not\subset t^*$, $\forall\, t^*$, $t^* \in R(-)$ (the equivalent condition is $s(t) \subseteq S(+)$).

Let k be the name of a set $R(k)$ of examples. To say that a collection t of values is a diagnostic test for $R(k)$ is equivalent to say that it does not cover any example t^*, $t^* \notin R(k)$. At the same time, the condition $s(t) \subseteq S(k)$ implies that the following implicative dependency is true: 'if t, then k'. Thus a diagnostic test, as a collection of values, makes up the left side of a rule of the first type.

It is clear that the set of all diagnostic tests for a given set $R(+)$ of examples (call it 'DT(+)') is the set of all the collections t of values for which the condition $s(t) \subseteq S(+)$ is true. For any pair of diagnostic tests t_i, t_j from DT(+), only one of the following relations is true: $s(t_i) \subseteq s(t_j)$, $s(t_i) \supseteq s(t_j)$, $s(t_i) \approx s(t_j)$, where the last relation means that $s(t_i)$ and $s(t_j)$ are incomparable, i.e. $s(t_i) \not\subset s(t_j)$ and $s(t_j) \not\subset s(t_i)$. This consideration leads to the concept of a good diagnostic test.

Definition 2. A collection $t \subseteq T$ ($s(t) \neq \varnothing$) of values is a good test for the set $R(+)$ of examples if and only if $s(t) \subseteq S(+)$ and simultaneously the condition $s(t) \subset s(t^*) \subseteq S(+)$ is not satisfied for any t^*, $t^* \subseteq T$, such that $t^* \neq t$.

Now we shall give the following definitions.

Definition 3. A collection t of values is irredundant if for any value $v \in t$ the following condition is satisfied: $s(t) \subset s(t/v)$.

If a collection t of values is a good test for $R(+)$ and, simultaneously, it is an irredundant collection of values, then any proper subset of t is not a test for $R(+)$.

Definition 4. A collection $t \subseteq T$ of values is maximally redundant if for any implicative dependency $X \rightarrow v$, which is satisfied in R, the fact that t contains X implies that t also contains v.

If t is a maximally redundant collection of values, then for any value $v \notin t$, $v \in T$ the following condition is satisfied: $s(t) \supset s(t \cup v)$. In other words, a maximally redundant collection t of values covers the number of examples greater than any collection $(t \cup v)$ of values, where $v \notin t$.

If a diagnostic test t for a given set $R(+)$ of examples is a good one and it is a maximally redundant collection of values, then for any value $v \notin t$, $v \in T$ the following condition is satisfied: $(t \cup v)$ is not a good test for $R(+)$.

Any example t in R is a maximally redundant collection of values because for any value $v \notin t$, $v \in T$ $s(t \cup v)$ is equal to $\varnothing$.

For example, in Table 1 the collection '*Blond Bleu*' is a good irredundant test for class 1 and simultaneously it is maximally redundant collection of values. The collection '*Blond Embrown*' is a test for class 2 but it is not good and simultaneously it is maximally redundant collection of values.

The collection '*Embrown*' is a good irredundant test for class 2. The collection '*Red*' is a good irredundant test for class 1. The collection '*Tall Red Bleu*' is a good maximally redundant test for class 1.

It is clear that the best tests for pattern recognition problems must be **good irredundant tests**. These tests allow constructing the shortest rules of the first type with the highest degree of generalization.

Table 1. Example 1of Data Classification (This example is adopted from (Ganascia, 1989))

Index of example	Height	Color of hair	Color of eyes	Class
1	Low	Blond	Blue	1
2	Low	Brown	Blue	2
3	Tall	Brown	Embrown	2
4	Tall	Blond	Embrown	2
5	Tall	Brown	Blue	2
6	Low	Blond	Embrown	2
7	Tall	Red	Blue	1
8	Tall	Blond	Blue	1

One of the possible ways for searching for good irredundant tests for a given class of positive examples is the following: first, find all good maximally redundant tests; second, for each good maximally redundant test, find all good irredundant tests contained in it. This is a convenient strategy as each good irredundant test belongs to one and only one good maximally redundant test with the same interpretation (Naidenova, 1999).

Inductive Rules of the Second Type

We use the lattice theory as the mathematical model for constructing good classification tests. We define a diagnostic test as a dual object (Naidenova, 2001), i. e. as an element of the concept lattice introduced in the Formal Concept Analysis (Wille, 1992).

The links between dual elements of concept lattice reflect both inclusion relations between concepts (structural knowledge) and implicative relations between concept descriptions (deductive knowledge).

Inferring the chains of lattice elements ordered by the inclusion relation lies in the foundation of generating all types of diagnostic tests. We use the following variants of inductive transition from one element of a chain to its nearest element in the lattice:

(i) from $s_q = (i_1, i_2, \ldots, i_q)$ to $s_{q+1} = (i_1, i_2, \ldots, i_{q+1})$;

(ii) from $t_q = (A_1, A_2, \ldots, A_q)$ to $t_{q+1} = (A_1, A_2, \ldots, A_{q+1})$;

(iii) from $s_q = (i_1, i_2, \ldots, i_q)$ to $s_{q-1} = (i_1, i_2, \ldots, i_{q-1})$;

(iv) from $t_q = (A_1, A_2, \ldots, A_q)$ to $t_{q-1} = (A_1, A_2, \ldots, A_{q-1})$.

We have constructed the special rules for realizing these inductive transitions: the generalization rule, the specification rule, the inductive diagnostic rule, and the dual inductive diagnostic rule.

The Generalization Rule

The generalization rule is used to get all the collections of indices $s_{q+1} = \{i_1, i_2, \ldots i_q, i_{q+1}\}$ from a collection $s_q = \{i_1, i_2, \ldots i_q\}$ such that $t(s_q)$ and $t(s_{q+1})$ are tests for a given class of positive examples.

The termination condition for constructing a chain of generalizations is: for all the extension s_{q+1} of s_q, $t(s_{q+1})$ is not a test for a given class of positive examples.

The Specification Rule

The specification rule is used to get all the collections of values $t_{q+1} = \{A_1, A_2, \ldots, A_{q+1}\}$ from a collection $t_q = \{A_1, A_2, \ldots, A_q\}$ such that t_q and t_{q+1} are irredundant collections of values and they are not tests for a given set of positive examples.

The termination condition for constructing a chain of specifications is: for all the extensions t_{q+1} of t_q, t_{q+1} is either a redundant collection of values or a test for a given set of positive examples.

The Inductive Diagnostic Rule

The inductive diagnostic rule is used to get a collection of values $t_{q+1} = \{A_1, A_2, \ldots, A_{q+1}\}$ from a collection $t_q = \{A_1, A_2, \ldots A_q\}$ such that t_q is not a test but t_{q+1} is a test for a given set of positive examples.

We extend t_q by choosing values which appear simultaneously with it in the examples of $R(+)$ and do not appear in any example of $R(-)$. These values are to be said essential ones.

The Dual Inductive DiagnostIc Rule

The dual inductive diagnostic rule is used to get a collection of indices $s_{q-1} = (i_1, i_2, \ldots, i_{q-1})$ from a collection $s_q = (i_1, i_2, \ldots, i_q)$ such that $t(s_{q-1})$ is a test but $t(s_q)$ is not a test for a given set of positive examples. This rule uses a method for choosing indices admissible for deleting from s_q. By analogy with an essential value, we define an essential example (Naidenova, 2005).

The rules for constructing diagnostic tests as elements of dual lattice generate logical rules of the first type, as shown in Table 2.

Table 2. Deductive Rules of the First Type Obtained with the Use of Inductive Rules for Inferring Diagnostic Tests

Inductive rules	Action	Inferring deductive rules of the first type
Generalization rule	Extending *s* (narrowing *t*)	Implications
Specification rule	Extending *t* (narrowing *s*)	Implications
Inductive diagnostic rule	Searching for essential values	Diagnostic rules
Dual inductive diagnostic rule	Searching for essential examples	Compatibility rules or (approximate implications)

The analysis of the inference for lattice construction allows demonstrating that this inference engages both inductive and deductive reasoning rules of the second type.

Both the generalization and specification rules realize the joint method of similarity-distinction. The extending of *s* results in obtaining the subsets of positive examples of more and more power with more and more generalized features (set of values). An algorithm NIAGaRa based on this variant of generalization rule is used in (Naidenova, 2006), for inferring good maximal redundant tests.

Both the inductive and dual inductive diagnostic rules are based on the inductive method of only distinction.

For example, a variant of the generalization rule involves the following deductive and inductive reasoning rules of the second type: the joint method of similarity-distinction, the rule of using forbidden rules of the first type, the method of only similarity, the rule of using implication, lattice operations.

It is important to note that the rules of the first type (implications, interdictions, rules of compatibility) generated during the lattice construction used immediately in this process.

Conclusion

This work is an attempt to transform a large class of machine learning tasks into a commonsense reasoning process based on using well-known deduction and induction reasoning rules. The key concept of our approach is the concept of a good diagnostic test. We have used the lattice theory as the mathematical model for constructing good diagnostic tests for learning implications from examples.

We have divided commonsense reasoning rules in two classes: rules of the first type and rules of the second type. The rules of the first type are represented with the use of implicative logical statements. The rules of the second type or reasoning rules (deductive and inductive) are rules with the help of which rules of the first type used, updated and inferred from data.

The analysis of the inference for lattice construction allows demonstrating that this inference engages both inductive and deductive reasoning rules of the second type.

Bibliography

[Cosmadakis et al., 1986] S. Cosmadakis, P. C. Kanellakis, N. Spyratos, "Partition Semantics for Relations", *Journal of Computer and System Sciences*, Vol. 33, No. 2, pp.203-233, 1986.

[Ericson et al., 1992] H. Ericson, A. R. Puerta, and M. A. Musen, "Generation of Knowledge Acquisition Tools from Domain Ontologies", International Journal of Human Computer Studies, Vol. 41, pp. 425-453, 1992.

[Ganascia, 1989] J.- Gabriel. Ganascia, "EKAW - 89 Tutorial Notes: Machine Learning", *Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Paris, France, pp. 287-296, 1989.

[Gappa and Poeck, 1992.] U. Gappa, and K. Poeck, "Common Ground and Differences of the KADS and Strong Problem Solving Shell Approach", EKAW – 92, Lecture Notes in Artificial Intelligence, No. 599, pp. 52-73, 1992.

[Lichko, 1983.] A A. E. Lichko, Psychopathies and Accentuations of Character of Teenagers, second edition, Leningrad, USSR, "Medicine", 1983.

[Mill, 1900] J. S. Mill, *The System of Logic*, Russian Publishing Company "Book Affair": Moscow, Russia, 1900.

[Naidenova, 1996] X. A. Naidenova, "Reducing Machine Learning Tasks to the Approximation of a Given Classification on a Given Set of Examples", *Proceedings of the 5-th National Conference at Artificial Intelligence*, Kazan, Tatarstan, Vol. 1, pp. 275-279, 1996.

[Naidenova, 1999] X. A. Naidenova, "The Data-Knowledge Transformation", in: "*Text Procesing and Cognitive Technologies", Paper Collection*, editor Solovyev, V. D., - Pushchino, Russia, Vol. 3, pp. 130-151, 1999.

[Naidenova, 2001] X. A. Naidenova, "Inferring Good Diagnostic Tests as a Model of Common Sense Reasoning", *Proceedings of the International Conference "Knowledge-Dialog-Solution" (KDS'2001)*, State North-West Technical University, Publishing House «Lan», Saint-Petersburg, Russia, Vol. II, pp. 501-506, 2001.

[Naidenova, 2005] X. A. Naidenova, "DIAGARA: an Incremental Igorithm for Inferring Implicative Rules from Examples", International Journal " Information Theories & Applications", Vol. 12, pp. 171-186, 2005.

[Naidenova, 2006] X. A. Naidenova, "An Incremental Learning Algorithm for Inferring Logical Rules from Examples in the Framework of the Common Reasoning Process", in: "*Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques",* editors Evangelos Triantaphyllou and Giovanni Felici*,* Springer*, pp. 90-146, 2006.*

[Naidenova and Polegaeva, 1985a] X. A. Naidenova, J. G. Polegaeva, "A Model of Human Reasoning for Deciphering Forest's Images and its Implementation on Computer", Theses of Papers and Reports of School-seminar "Semiotic Aspects of the Formalization of Intellectual Activity", Kutaisy, Georgia Soviet Socialist Republic, pp. 49-52, 1985a.

[Naidenova and Polegaeva, 1985b] X. A. Naidenova, J. G. Polegaeva, "The Project of Expert System GID KLARA – Geological Interpretation of Data Based on Classification and Pattern Recognition", Report I-A VIII.2 10-3/35, "Testing and Mastering Experimental Patterns of Flying (Aircraft) and Surface Spectrometry Apparatus, Working out Methods of Automated Processing Multi-Spectral Information for Geological Goals", All Union Scientific Research Institute of Remote Sensing Methods for Geology, 1985b.

[Naidenova and Syrbu, 1984] X. A. Naidenova, V. N. Syrbu, "Classification and Pattern Recognition Logic in Connection with the Problem of Forming and Using Knowledge in Expert Systems", Theses of Papers of Republican Scientific-Technical Conference "Interactive Systems and Their Practical Application", Mathematical Institute with Computer Center, Kishinev, Moldavia, pp. 10-13, 1984.

[Naidenova and Polegaeva, 1986] X. A. Naidenova, J. G. Polegaeva, "An Algorithm of Finding the Best Diagnostic Tests", *The 4-th All Union Conference "Application of Mathematical Logic Methods"*, Theses of Papers, Mintz, G; E, Lorents, P. P. (Eds), Institute of Cybernetics, National Acad. of Sciences of Estonia, Tallinn, Estonia, pp. 63-67, 1986.

[Wille, 1992] R. Wille, "Concept Lattices and Conceptual Knowledge System", *Computer Math. Appl.*, Vol. 23, No. 6-9, pp. 493-515, 1992.

Author's Information

Naidenova Xenia Alexandrovna - *Military medical academy, Saint-Petersburg, Stoikosty street, 26-1-248,* naidenovaxen@gmail.com naidenova@mail.spbnit.ru.

AN INTELLIGENT SYSTEM FOR INVESTIGATIONS AND PROVISION OF SAFETY FOR COMPLEX CONSTRUCTIONS

Alexander Berman, Olga Nikolaychuk, Alexander Yurin, Alexander Pavlov

***Abstract**: Methodology of computer-aided investigation and provision of safety for complex constructions and a prototype of the intelligent applied system, which implements it, are considered. The methodology is determined by the model of the object under scrutiny, by the structure and functions of investigation of safety as well as by a set of research methods. The methods are based on the technologies of object-oriented databases, expert systems and on the mathematical modeling. The intelligent system's prototype represents component software, which provides for support of decision making in the process of safety investigations and investigation of the cause of failure. Support of decision making is executed by analogy, by determined search for the precedents (cases) with respect to predicted (on the stage of design) and observed (on the stage of exploitation) parameters of the damage, destruction and malfunction of a complex hazardous construction.*

***Keywords**: computer-aided investigations, intelligent system, technical state, safety, construction, malfunction, failure, case-based reasoning.*

***ACM Classification Keywords:** I.2.1 Applications and Expert Systems: Medicine and science, Industrial automation*

Introduction

Prevention of failures in industry necessitates solution of the problem of investigation and provision of technogenic safety on all the stages of the life cycle of complex constructions: beginning from the design, construction engineering, manufacture and ending with application and utilization. The problem of safety investigations is a multi-disciplinary one. For the purpose of its solution it necessitates that knowledge and the potential of the following scientific disciples be involved: physics and mechanics of destruction, physics-chemical mechanics of materials, material engineering, reliability and safety of technologies and constructions, toxicology, foundations of design, technology of mechanical engineering, psychology, mathematics, information technologies, etc. Safety is substantially dependent on the efficiency of the systems intended for estimation and forecasting of the technical state and resource, precision of diagnostics and correct determination of the causes safety violations.

Investigation of incidents and failures, which implies description of the total cause-effect complex of their formation, is one of the main sources for acquisition of knowledge about hazards and their development [Berman, 1998].

In connection with the multi-disciplinary character of the problem of investigation and amplification of construction safety, it is necessary to provide for a coordinated activity of the researchers and the specialists, who regulate different stages of the complex systems' life cycle. This may be achieved only via elaboration of the respective computer-aided technologies intended for automation of research, which is conducted within the frames of an integrated intelligent information system and on the basis of accumulation, modeling, initial processing and efficient application of diverse information and knowledge [Berman at all, 1999].

Methodology of information support and automation of research related to technogenic safety

The methodology of computer-aided research and provision of safety properties is determined by the object's model, by the structure and functions of the process of investigation, and by the set of methods employed in the investigation.

Object's Model. Investigation and development of the recommendations related to provision of safety is based on identification and application of the regularities of the genesis, generation and development of the hazards independently of the functions and the structure of construction.

Correct determination and prediction of the causes allow us to make the objects more perfect, ground some necessary modernization, redefine the periodicity, the methods and aids needed for diagnostics and monitoring, ground the undertakings related to prevention of failures and provision of safety in case of their occurrence.

To the end of investigation of the causes of occurrence of hazardous states, we have proposed a cause-and-effect complex which determines their occurrence in the form of a model of dynamics of undesired processes. A

block-diagram of dynamics of this process is shown in Fig.1. Each sequential state is conditioned by the previous one and is characterized by the some larger hazard. So, since presently our knowledge bound up with understanding sufficiency of the measures and undertakings needed for provision of reliability and safety is limited and the systems, which are intended to maintain reliability of operation and safety, are hardly ever fail-safe, malfunctions of such systems take place and provoke failures.

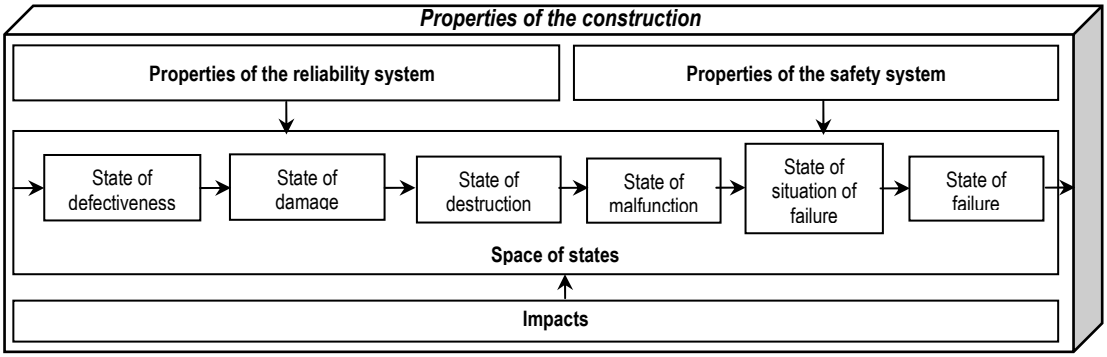


Fig.1. The cause-and-effect complex of state dynamics for the construction

According to this model, hazardous states are conditioned by properties of the designed system, by the character of impacts and by properties of systems intended to provide for reliability and safety. System's properties are characterized, for example, by i) the number of new system's components and ii) the degree of their uncertainty, iii) predictability of development of the states, iv) the level of hazard for the substances either processed or transported, v) the degree of hazard for the technological process parameters. The properties of impacts include those of mechanical, physical-chemical and biological influences, which violate safety; the rate of their development and distribution; the degree of their effect on the safe state. Properties of the safety system include, for example, observability; controllability; opportunity of state monitoring, which implies real-time information processing and realization of adequate measures for planning hazards; survivability, e.g. the operating time from the moment of occurrence of a hazardous state to the moment of transition onto another level of the hazardous state.

The structure and functions of the process of investigation. The process of investigation has a hierarchical structure. The structure of the process of investigation is conditioned by the following factors:

- the structure of the object under scrutiny: part – unit of an assembly – construction;
- the proposed structure of the state space: defect – damage – destruction – malfunction – situation of failure – failure [Berman, 1998; Berman at all, 2007];
- a set of mechanisms of occurrence and a variety of hazards which are the causes of safety violation;
- a set of scenarios bound up with development of each hazard;
- a set of variants of decisions bound up with provision of safety properties which satisfy the conditions of an acceptable risk.

The proposed structure of the model of the cause-and-effect complex is the decisive factor defining the scheme of investigation which includes consecutive investigation stages concerned with all the phases of states – from the appearance of a defect to the formation of failure. On each stage the factors are revealed, which condition and influence the frequency and consequences of hazardous states. This is necessary for the purpose of determination of the construction's rational preventive, control and protective properties, which are generalized in the concept of "property of safety". Methods and aids of provision of these properties are based on the results of such investigations.

Functions of the process of investigations correspond to the stages of decision making needed for achievement of the objectives bound up with provision of acceptable risk for all the kinds of hazardous states of constructions (Fig.2).

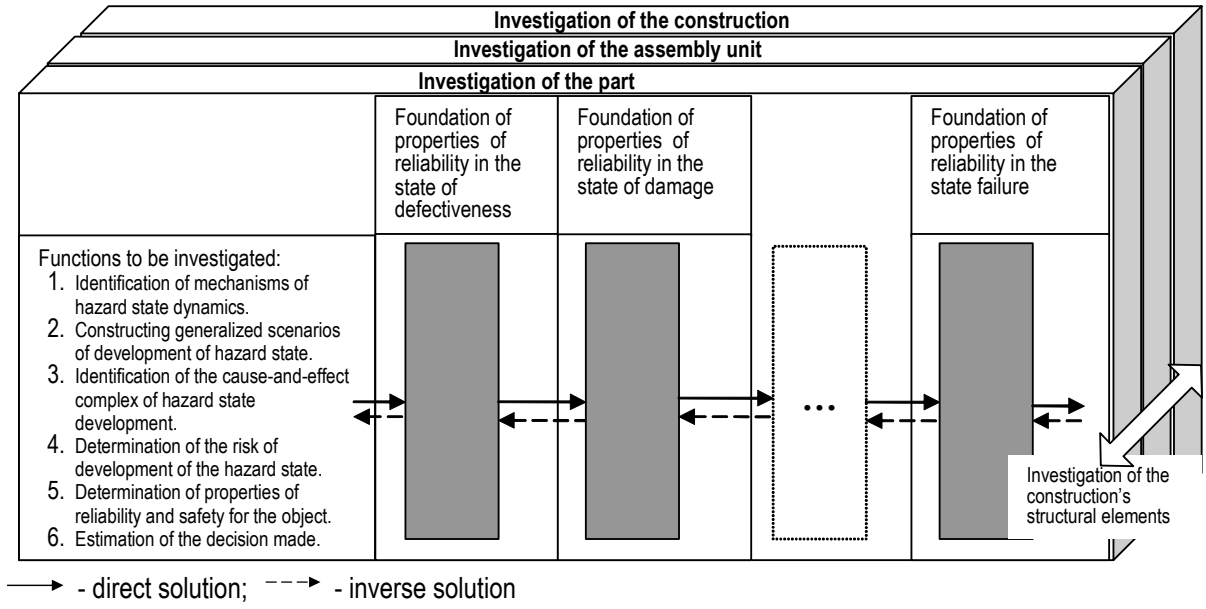


Fig.2. The structure of the process of investigation.

Methods of Investigation. In accordance with the methodology proposed, the process of investigation and provision and maintenance of safety for complex constructions is conducted with the aid of a set of methods based on the technologies of object-oriented databases, expert systems and mathematical modeling.

The process of investigation is provided by combining the methods (see Fig.3): case-based and rule-based expert systems, ontologies, methods of analytical modeling [Aamodt at all, 1994; Berman at all, 2004; Luger, 2002; Portinale at all, 2004]. In turn, before applying above methods it is necessary to perform modeling (redefining, constructing a thesaurus) of the problem domain.

Ontological systems belong to the most contemporary forms of information (data and knowledge) representation (Berman at al., 2004b). The principal intention of the ontology implies formalization and integration of information. Ontology facilitates structuring and modeling weakly-formalized problem domains. Being grounded on the general set of terms, it determines and simplifies the semantics of formal information, facilitates its computer processing, while representing the information in the form convenient from the viewpoint of perception.

Application of the mechanism of ontology in problems of providing safety of constructions is conditioned by insufficient formalization and multi-disciplinary character of the problem under scrutiny. Its solution necessitates application of knowledge in science of materials, solid body physics, physics and mechanics of destruction, physical and chemical mechanics and strength of materials, monitoring, diagnostics and forecasting, theories of risk and safety. Furthermore, likewise in all multidisciplinary investigations, there exists the problem of knowledge coordination and development of a uniform conceptual apparatus which would provide for efficient interaction between the researchers involved in different fields of knowledge.

So, first of all, it is necessary to construct an ontology (a dictionary-type ontology) of construction safety and then formalize the main concepts from the viewpoint of the cause-and-effect complex of safety violation.

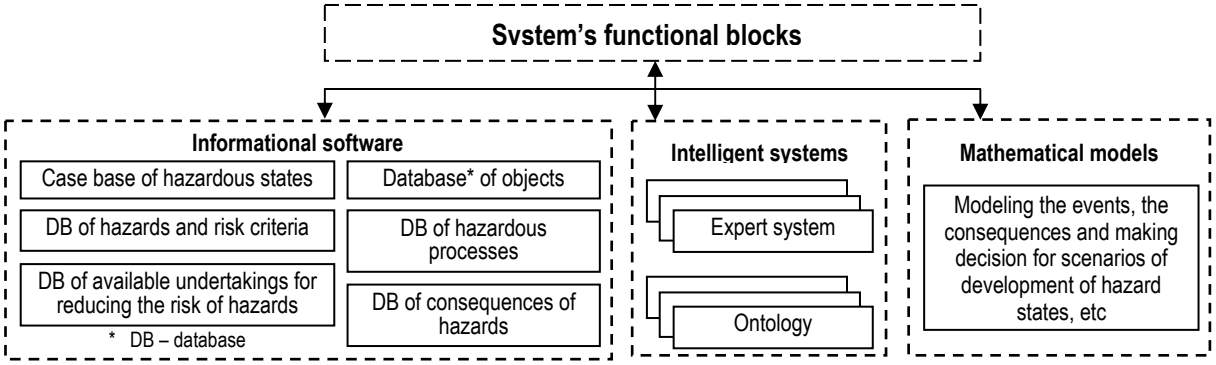


Fig.3. The architecture of the system for investigation and provision of reliability and safety of the constructions.

The formalized concepts form a taxonomy of concepts which inherit the properties of general concepts. Such abstract concepts as defect, damage, destruction, malfunction, situation of failure, failure have been decomposed into definite concepts. The ontology elaborated is supposed to be used as a knowledge base for the system proving for safety.

Case-based reasoning provides for solving the problems on the basis of precedents (cases), while using accumulated experience, i.e. the decisions made earlier [Aamodt at all, 1994]. According to this methodology, the process of solving the problem represents a sequence of stages related to finding (retrieval) some analogs and reuse of the information contained in them.

Furthermore, knowledge (apparatus) of different scientific and engineering disciplines is concentrated with respect to these precedents. So, each precedent represents some systematized and classified information on the causes of possible or actual damages and destruction of the construction, which condition either probable or actual (real) violation of safety (failure).

The precedent includes the following information: project and actual exploitation conditions; a sequence of the states, which could lead to the state under scrutiny, while including the properties of these states; causes of the state; the organizing-technical genesis of the state; the structural genesis of the state; consequences of the state; the decision made to prevent the undesirable state; etc.

The construction has a hierarchical structure, each element of which is characterized by a state. The relation "part–total" between the structural elements conditions the cause-effect relations between their technical states: a part, which represents an element of an assembly unit, may be the cause of the undesirable state for the assembly unit; etc.

Proceeding from this assumption, it is possible to state that there exists a hierarchical space of precedents. Furthermore, each precedent (case) corresponds to a set of indices, which represent a brief description of one of the declarative aspects of the precedent: for example, either structural belonging of the object (involved in the incident) or its technical state. Depending on the complexity of description (some available hierarchy of properties and a type of criteria: determined/logical) the indices (descriptors) represent either binary sequences (…01001…) or some sets of corteges ({..., P_i , ...}, $P_i = \langle n, v, w, r \rangle$, where n is the property's name; v is its value; w is the importance (or information weight) of the property; r is the restriction imposed on the band of values – the restriction determines the band of values within the frames of which the property's value can determine the value of the measure of similarity; in the case when the property's value occurs outside of this band of values the value of the similarity measure is 0).

Presence of the given set of indices allows us to apply elements of the procedure of sequential solutions [Zhuravlev at all, 1989] in the process of finding solution. Finding (retrieval) of the precedents with respect to separate indices and groups of indices (from the set of indices) allows one not only to substantially increase the search algorithm's computational power (and, therefore, complexity of the process of investigations) at the expense of restricting the number of vain (irrelevant) comparison and search operation, but also to concentrate attention of the researcher on some important aspects of the technical state dynamics.

Selection of precedents on each of the stages in the procedure of sequential solutions is conducted in accordance with a global measure (estimate) of similarity/closeness of descriptions of the precedents. This measure is computed as a distance between the precedents in the space of criteria (features). The distance is computed using both the Minkovsky metrics [Bergmann, 2002] (1), which is a generalization of the so called "city district metrics" (which is used in processing of binary vectors) and the Euclidean distance (used in processing some sets of corteges):

$$dist_{Minkowsi,p}(\bar{x},\bar{y}) = (\sum_{i=1}^{n}\left|x_i - y_i\right|^p)^{1/p} \quad (1)$$

The parameter p determines whether it behaves like the so called "city district metrics"($p=1$) or like the Euclidean distance.

The search within a hierarchical space may be both ascending (bottom – up) (i.e. from a part to a construction) and descending (top to bottom). In the first case, the problem of forecasting is solved in the process of investigation; in the second cease, the problem of genesis is solved.

Consider the sequence of stages (steps) in the search algorithm intended for finding precedents in the process of solving the problem of genesis. This problem can be solved on the stage of exploitation of the object, when some malfunction of the construction has already taken place, external manifestations of the malfunction are obvious, and it is necessary to find out the causes of this malfunction.

Step 1. The user describes the object of investigation, i.e. the failed construction, while taking into account initial exploitation conditions (values) and external manifestations of malfunction, and then automatically forming the initial set of indices needed for the description.

Next, search for the analogs with respect to the indices, which describe structural belonging of the failed construction and external manifestations of malfunction is sequentially conducted. The result of the procedure of finding (retrieval) is a list of similar precedents ordered according to the estimate of closeness (similarity) of the descriptions. This list allows one to analyze the malfunctions having similar indicators and make a preliminary decision related to investigation of causes of the malfunction, for example, a decision on testing all the assembly units (or parts), which have been indicated in the list as possible causes of malfunction.

Step 2. Having obtained additional information in the form of a list of similar construction malfunctions, the user continues the process of investigation – determines the genesis of the malfunction. The structural genesis of malfunctions of all constructions is the malfunction of one of the assembly units (or parts) included in some or another construction. A set of possible structural genesises of construction's malfunctions is formed in the process of investigation of these assembly units (parts). The procedure of finding (retrieval) proceeds to the next level in the hierarchy of the space of precedents (cases), to the level of precedents describing the malfunctions of assembly units (or parts).

In turn, already on the given level, there takes place the search of analogs with respect to external manifestations of some malfunction, the result of which are some precedents containing the description of the most probable cause of the malfunction for the assembly unit. Having chosen the most close description of the malfunction (and, consequently, the most probable failed assembly unit), the user redefines the actual conditions of its exploitation and external manifestations of its malfunction.

Step 3. Since the object of investigation represents itself a hierarchical structure, and the process of investigation represents a sequential and goal-oriented search of analogs in the hierarchical space of precedents, after finding

the analogs at the level of precedents, which describe malfunctions of assembly units, there follows the search for the precedents at the next level, i.e. at the level of precedents, which describes malfunctions of parts. At the given level, the search (retrieval) of analogs is conducted with respect to the indices, which describe external manifestations of damage, destruction and malfunction of the part, and on account of information of actual conditions of its exploitation and initial defectiveness (conditioned by the technology of manufacture). Such a description of this step is given in [Nikolaychuk at al, 2006].

Only on the given step the user reaches the beginning of the chain of the structural genesis of malfunction and obtains the possibility to "assign" and analog, i.e. to choose the most close precedent. As a result of this "assignment", attribution of the solution, which is contained in the analog, to a new precedent, which characterizes the current situation, takes place. The solution contains a description of the organizing-technical causes of malfunction for the part as well as a description of undertakings, which are needed to prevent any malfunctions of the part in future.

Step 4. As soon as the cause of the part's malfunction has been determined, the algorithm turns back to the stage of finding out the malfunction for the assembly unit. On this stage, the organizing-technical genesis of malfunction of the part is inherited by the assembly unit. The part in the structural genesis is fixed, which caused the malfunction of the assembly unit. Noteworthy, after "assigning" the analog, we obtain the total chain of the cause-effect complex in the form of a set of precedents: the part's precedent – the assembly unit's precedent – the construction's precedent. On each of the levels, the precedent contains information on the cause of malfunction and the undertakings needed. So, the undertakings, which need to be conducted for the assembly unit, are inherited from a similar precedent of malfunction for some assembly unit and are redefined by the user on account of the definite characteristics of the problematic situation.

Step 5. As soon as the cause of the assembly unit's malfunction has been determined, the algorithm turns back to the stage of finding out the malfunction for the construction on the whole. On this stage, the organizing-technical cause of malfunction of the assembly unit is inherited by the construction. The assembly unit in the structural genesis is fixed, which caused the malfunction of the construction. The undertakings, which need to be conducted for the construction, are inherited from a similar precedent of malfunction for some other construction and are redefined by the user on account of the definite characteristics of the problematic situation.

Therefore, it appears possible to determine (trace) the whole chain of genesis for the malfunction. Before the moment of "assigning" an analog, the user can turn back to the previous step, input additional information and repeat the search (retrieval) of analogs.

Application of analogs in many cases necessitates adaptation of available solutions, what may be implemented by the concretization, which implies qualitative redefining the description of the precedent, refining the parameters and their values.

The problem of genesis of states for a failure situation or a failure is solved by the case-based method likewise in the problem of genesis of construction malfunction considered above.

The rule-based reasoning provides for solving problems of investigation on the basis of models of the cause-effect complex of occurrence of malfunctions and failures. A corresponding rule-based model has been developed, where the object concepts and relations between them have been transformed into the production rules of CLIPS [Nikolaychuk at all, 2006].

The mathematical models. In the problem domain under scrutiny, side by side with weakly formalized knowledge, there are separate aspects of the technical state dynamics which are described by analytical models, for example, by models of growth of micro-cracks, variation of material hardness, variation of residual stress (strains) in the part, etc. So, in the process of describing the process of variation of the technical state, it is necessary to combine rule-based models (and/or case-based model) and analytical models, when the latter complement and redefine the values of separate parameters of these knowledge models.

Application (Intelligent System)

Implementation of the system is conducted by component-wise assembly of the systems designed by the team [Berman at all, 2006]. Each component represents an autonomous module having some internal memory and a unified interface. The internal memory provides the user with the opportunity to input some information needed for modification of the basic functionality, its adjustment to the specificity of a definite problem domain. The unified interface represents a set of properties and the methods needed for obtaining a description of a component and for controlling it.

In the process of design of these components it was necessary to isolate the employed knowledge of some problem domain from the knowledge of the object domain; the component has to know "how" the information is processed, but not "what" is definitely processed. When some purchased software is employed, the implementation of a component consists in design of a controlling module for this software. The unified interface gives the possibility of programmed control of a component, what provides for the possibility of dynamic integration of the system's components into a joint system.

Presently, we have elaborated a prototype of intelligent system intended for determination of causes of malfunctions and failures in the oil-chemical industry [Berman at all, 2006], which includes the following components: component for modeling (description) of an object domain, which provides for integration of the information on the object domain [Berman at all, 2004a]; case-based and rule-based expert systems [Nikolaychuk at all, 2006]; modules, which implement mathematical functions. Furthermore, there are databases, which contain information on the constructions, degradation processes, consequences and undertakings oriented to neutralization of the impact. The knowledge base of the case-based expert system contains information on 250 malfunctions and failures which have taken place at different oil-chemical enterprises. The rule base of the rule-based expert system includes the rules for the relationships between degradation processes, causes and the related undertakings. Mathematical models intended for computing the rate of cracking development and corrosion have been developed.

Conclusion

Provision of predicted technical state, estimation of the possible hazard of the object's destruction and the expected damages (detriment) acquires special importance at the stages of design of technical devices intended to be applied under special, extreme conditions, for example, under water or in space. It acquires importance also in connection with the necessity of all the more increased degree of automation and informatization of all the life cycle stages of complex constructions performed on the basis of adaptive and intelligent control systems.

Development of computer-aided systems of automation and informatization of research, which imply storage, initial processing, modeling and efficient application of diverse information and knowledge within the frames of one integrated intelligent information system, ensures coordinated activity of researchers and specialists in solving multi-disciplinary problems of investigation and increase of safety of the complex constructions on various stages of their life cycle.

Bibliography

[Aamodt at al, 1994] A. Aamodt, E Plaza. Case-Based reasoning: Foundational issues, methodological variations, and system approaches. In: *AI Communications*, 7(1994), No.1, 39-59.

[Bergmann, 2002] R. Bergmann. Experience Management. In: *Lecture notes on artificial intelligence*, 2432 (2002), 364.

[Berman, 1998] A.F. Berman. Degradation of Mechanical Systems. Novosibirsk: Nauka, 1998. (in Russian).

[Berman at al, 1999] A.F. Berman, O.A. Nikolaychuk. Structurization in investigations of safety for complex technical systems. In: *Safety problems of extreme situations*, 6 (1999), 3-14 (in Russia).

[Berman at al, 2004a] A.F. Berman, O.A. Nikolaychuk, A.I. Pavlov, A.Y. Yurin. Ontology of mechanical systems reliability. In: Artificial intelligence, 3 (2004), 266-271, Institute of Artificial Intelligence Press, Donetsk, Ukraine.

[Berman at al, 2006] A.F. Berman, O.A. Nikolaychuk, A.I. Pavlov, A.Y. Yurin. An intelligent system for support of decision making in the process of determination of causes of malfunctions and failures in the oil-chemical industry. In: *Automation in Industry*, 6 (2006), 15-17. (in Russia)

[Berman at al, 2007] A.F. Berman, O.A. Nikolaychuk. The space of technical states for unique mechanical systems. In: *Journal of Machinery Manufacture and Reliability* (*Problemy Mashinostroeniya i Nadezhnosti Mashin*), 1 (2007), 14-22.

[Feret at al, 1997] M.P. Feret, J.I. Glasgow. Combining Case-Based and Model-Based Reasoning for the Diagnosis of Complex Devices. In: *Applied Intelligence*, 7 (1997), 57–78.

[Luger, 2002] G. F. Luger Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 5th Edition, Addison-Wesley, 2002.

[Nikolaychuk at al, 2006] O.A. Nikolaychuk, A.Y. Yurin. The prototype of an intelligent system for computer-aided investigation of a technical state for unique machines and constructions. In: *Artificial intelligence*, 4 (2006), 459-468, Institute of Artificial Intelligence Press, Donetsk, Ukraine.

[Portinale at al, 2004] L. Portinale, D. Magro, P.Torasso, Multi-modal diagnosis combining case-based and model-based reasoning: a formal and experimental analysis. In: *Artificial Intelligence*, 158 (2004), no.2, 109-154.

[Zhuravlev at al, 1989] Zhuravlev, I. Yu., & Gurevitch, I. B. (1989). Pattern recognition and image recognition. In Yu. I. Zhuravlev (Ed.), *Pattern recognition, classification, forecasting: Mathematical techniques and their application*, issue 2 (5–72). Moscow: Nauka. (in Russia)

Authors' Information

Alexander Berman – *Institute for Systems Dynamics and Control Theory, Russian Academy of Sciences, Siberian Branch (ISDCT, SB of RAS), Head of Laboratory; Laboratory for Methods of Automation of Technogenic Safety Investigations; Box 664033, Lermontov st., Irkutsk, Russia; e-mail: berman@icc.ru*

Olga Nikolaychuk - *ISDCT RAS SB, senior researcher; e-mail: nikoly@icc.ru*

Alexander Pavlov - *ISDCT RAS SB, researcher; e-mail: Asd@icc.ru*

Alexander Yurin - *ISDCT RAS SB, researcher; e-mail: iskander@icc.ru*

ЭКСПЕРТНАЯ СИСТЕМА КОНТРОЛЯ ОРГАНОЛЕПТИЧЕСКИХ ПОКАЗАТЕЛЕЙ КАЧЕСТВА МЯСНОЙ ПРОДУКЦИИ

Вадим Зайцев

Abstract: *In paper the modern condition of the control of manufacture and quality of meat production is submitted. The original method of allocation of contours and structures of inclusions of various fractions on the images researched tests inspection production of a meat industry is offered*

Ключевые слова: *метод выделения контуров, автоматизированные системы контроля.*

Введение

В настоящее время в Украине качество мясной продукции, предлагаемой потребителю, значительно ухудшилось. Особенно резко понизилось качество колбасных изделий. По некоторым данным, мяса в большинстве видов современной колбасной продукции содержится не более 10 -15%.

Кроме того, иногда используется испорченные мясо или колбасы. На мясоперерабатывающих комбинатах эту испорченную продукцию подвергают обеззараживанию химическими реактивами и вторичной переработке.

Нарушение процесса производства колбасной продукции часто заключается в несоответствии этой продукции требованиям нормативных документов и техническим условиям. По результатам испытаний колбасных изделий, проведенных на торговых предприятиях Украины, на соответствие нормативной документации почти половина проверенной продукции не соответствовала норме. В продукции обнаружено завышенное содержание влаги, нитритов и фосфатов. Эти вещества добавляются в колбасные изделия с целью увеличения их веса, продления срока хранения и придания колбасам приятного розового цвета. Известно, что повышенное содержание в продукции нитратов и фосфатов весьма небезопасный фактор для здоровья человека [Женкова, 2003].

Поэтому контроль мясной продукции, является одним из важнейших факторов в защите интересов потребителя, так как от этого напрямую зависит здоровье граждан. Чем выше уровень контроля продовольственной продукции в стране, тем больше гарантии у потребителя, что приобретенный товар – действительно качественный и безопасный.

В Украине, в настоящее время, делу защиты прав потребителей и технического регулирования уделяется огромное внимание, Верховная Рада Украины приняла в первом чтении закон "О мясе и мясных продуктах", который определяет правовые, организационные основы обеспечения качества и безопасности мяса и мясопродуктов для жизни и здоровья населения [Мозговая, 2005].

В ГОСТах на производство колбасной продукции имеются четкие указатели состава той или иной продукции. Но эти показатели не соблюдаются.

Нет методики, с помощью которой можно было бы вычислить в колбасных изделиях количество белков растительного или животного происхождения [ГП"Севастопольстандартметрология,2004].

Основным механизмом по борьбе с фальсификацией является тщательный контроль качества потребительских товаров. Несоответствие продукции требованиям нормативной документации в среднем по ряду предприятий страны (завышение массовой доли влаги 55%, массовой доли нитритов 25%, несоответствие по органолептическим показателям 13%) указывает на необходимость ежедневных проверок продукции. В связи с этим, разработка и применение автоматизированных систем контроля качества мясной продукции весьма актуальны и своевременны.

В соответствии с положениями ГОСТ 9959-91 органолептическая оценка проводится для определения соответствия органолептических показателей качества продуктов требованиям нормативно-технической документации, а также для оценки новых видов мясной продукции при постановке ее на производство. Органолептическая проверка осуществляется специалистами дегустаторами для определения показателей – внешнего вида, цвета, вкуса, аромата, консистенции и др. посредством органов чувств субъективно.

Результаты органолептической оценки сопоставляют с показателями качества, приведенными в нормативно-технической документации на данный вид продукта, определяя при этом соответствие продукта требованиям стандарта или технических условий.

Автоматизация процесса органолептической оценки качества мясной продукции

Для автоматизации органолептической оценки качества мясной продукции разработана экспертная система WurstControl.

Система разработана в среде Borland C++ Builder 5.0 под платформы Windows 95/98/NT/2000.

WurstControl решает задачу выделения органолептических признаков качества продукции по ее изображению исходя из базы данных эталонов – изображений проб продукции. Изображения исследуемых проб продукции подаются на вход системы посредством цифровой видеокамеры или из файла заранее записанных изображений, или из буфера обмена системы Windows.

На этапе предварительной обработки изображения проводится абстрагирование от цвета, освещенности и фонового шума. Для осуществления этой операции используется переход от растрового входного цветного изображения к представлению его в виде контурных линий.

Методы выделения контурных линий

Задача выделения контуров на изображении давно успешно решается классическими алгоритмами.

Анализ робот, посвященных выделению контуров областей изображений, позволяет выделить три основных метода [Дуда., Харт, 1976] , [Hall. 1979], [Fu K., Mu J., 1981]:

Пространственное дифференцирование;

Функциональная аппроксимация;

Высокочастотная фильтрация.

Метод пространственного дифференцирования

В этом методе используется алгоритм пространственного дифференцирования, который преобразует изображение в скалярное поле g(x, y),

$$g(x,y) = \left\| \nabla f(x,y) \right\|, \quad \forall x \in X \ ,$$

где

$$\left\| \nabla f(x,y) \right\| = \left\{ \left[\frac{\partial f(x,y)}{\partial x} \right]^2 + \left[\frac{\partial f(x,y)}{\partial y} \right]^2 \right\}^{1/2} ,$$

$\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}$ - функции яркости изображения на ортогональных направлениях; X - область задания функции f(x, y). Поле g(x, y) называют градиентным изображением с усиленными границами [5-7]. Обработка градиентного изображения осуществляется с помощью граничного оператора

$$b(x,y) = \begin{cases} 1 \ \text{при} \ g(x,y) \ge T, \\ 0 \ \text{при} \ g(x,y) < T, \end{cases} \qquad (1)$$

где T – размер принятого порога.

В результате удается получить бинарное контурное изображение b(x,y), элементы которого определяют границы f(x, y).

Метод функциональной аппроксимации

Метод позволяет решить задачу выделения границ с помощью оптимизационных алгоритмов. Здесь для каждой точки изображения (x', y') рассматривается окружность с центром в точке R. Для элементов обозначенной окружности выделяется функция вида:

$$\hat{f}(x,y,c_1,c_2,t,a_1,a_2) = \begin{cases} a_1 & \text{при } c_1 x + c_2 y \ge t, \ (x,y) \in R, \\ a_1 + a_2 & \text{при } c_1 x + c_2 y < t, \ (x,y) \in R, \\ 0 & \text{при } (x,y) \notin R, \end{cases}$$

где c_1, c_2, t, a_1, a_2 - численные параметры.

Эта функция определяет "идеальный край" в некоторой точке (x', y'). Ориентация этого края и его место расположения относительно центра окружности определяется параметрами c_1, c_2, t, а амплитудные характеристики края - параметрами a_1, a_2. Решение задачи сводится к аппроксимации функции f(x, y). Качество аппроксимации оценивается метрикой $\rho(f, f)$ в пространстве функции, которая интегрируется в квадрате:

$$\rho(f,\hat{f}) = \iint_R \left[f(x,y) - \hat{f}(x,y,c_1,c_2,t,a_1,a_2) \right]^2 dxdy$$

Если удается подобрать параметры аппроксимируемой функции, которые обеспечивают заданное качество аппроксимации (размер ρ метрики), то принимается решение о наличии края в точке (x', y'). При этом стают известными его ориентация и амплитудные характеристики. Алгоритм известен так же, как оператор Хюккеля.

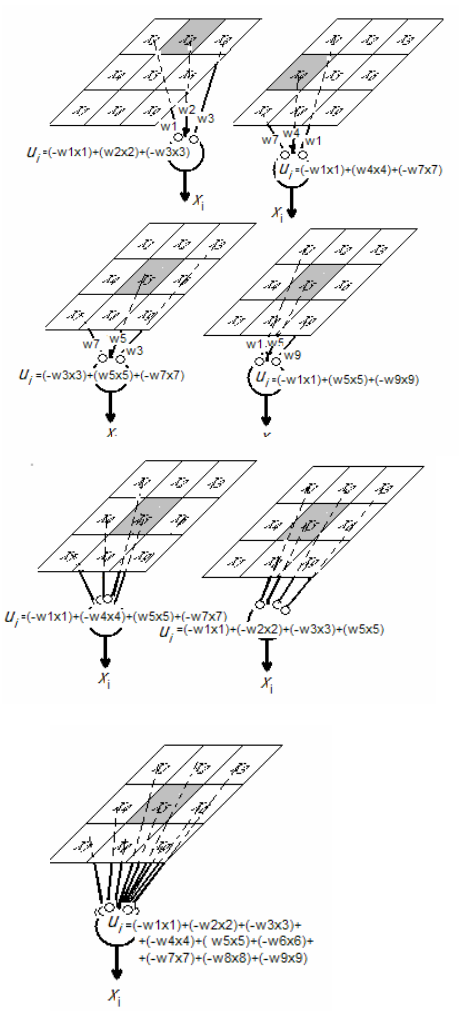


Рис.1. Определение переменной u_i

Метод высокочастотной фильтрации

Метод высокочастотной фильтрации позволяет решить задачу выделения границ с помощью обработки изображения в области пространственных частот. Этот метод основан на том. Что информацию о границах объектов несут высокочастотные составляющие спектра изображения [4,6].

Пусть $F(\varpi_x, \varpi_y)$ - спектр Фурье функции яркости изображения $f(x,y)$; $H(\varpi_x, \varpi_y)$ - передаточная характеристика высокочастотного фильтра; $FR^{-1}\{\cdot\}$- оператор двухмерного преобразования Фурье. Тогда соотношение

$$g'(x,y) = FR^{-1}\left\{F(\varpi_x,\varpi_y)H(\varpi_x,\varpi_y)\right\}$$

определяет изображение з подчеркнутыми резкими перепадами яркости. Далее к функции $g'(x,y)$ для получения конечного результата можно применить граничный оператор вида (1).

Общим для всех этих методов есть стремление рассматривать границу как область резкого перепада функции яркости изображения f(x,y). Отличает же их математическая модель понятия край и алгоритм поиска краевых точек [Дуда., Харт, 1976] , [Hall. 1979], [Fu K., Mu J., 1981].

Реализация каждого из трех методов осуществляется с помощью алгоритмов специальных классов, которые имеют разную вычислительную сложность и скорость, а также обладают различными требованиями к емкости оперативной памяти ЭВМ.

Наиболее громоздким есть метод высококачественной фильтрации. Для его алгоритмичной и программной реализации не обходимо применение алгоритмов быстрого двумерного преобразования Фурье и тщательный подбор передаточной функции фильтра. С другой стороны необходима большая емкость оперативной памяти ЭВМ при пофрагментной обработке изображения.

Метод функциональной аппроксимации также достаточно сложный. Решение оптимизационной задачи подбора параметров функции необходимо выполнять для каждой из N^2 точек изображения.

Таким образом, все эти методы имеют ряд недостатков, главные из которых – это сложность реализации, а также относительно низкая скорость работы, которой будет недостаточно для обработки большого количества изображений в режиме реального времени.

Для решения задачи выделения контуров структурных признаков продукции был применен бионический подход, в котором обработка информации производится посредством нейронной сети. Благодаря полному параллелизму обработки каждой точки изображения, при аппаратной реализации этого подхода, возможна высокая скорость обработки входных изображений.

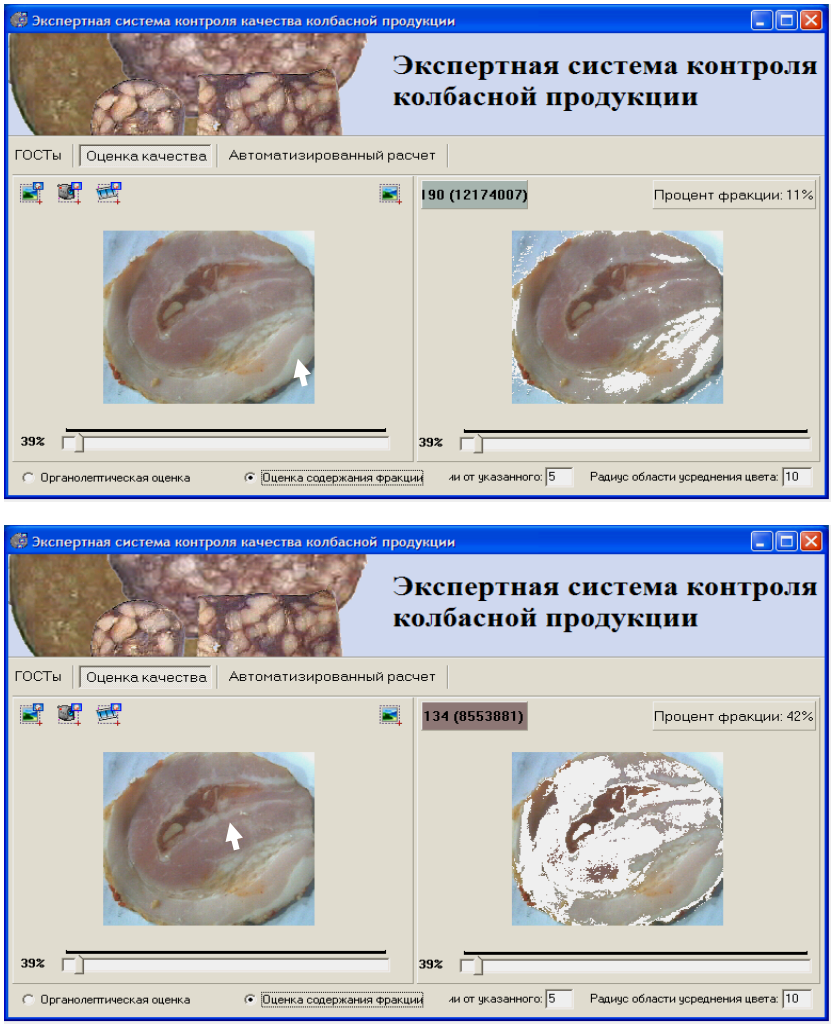


Рис.2. Выделение различных фракций продукции

Нейрометод выделения структурных признаков качества продукции на его изображении

В бионическом подходе считается, что каждая точка изображения отвечает одному нейронному элементу («нейрону»). Нейронный элемент состоит из сумматора входных сигналов Σ и преобразователя F с заданной, как правило, нелинейной характеристической (активационной функцией). Сумматор генерирует переменную $u_i = \sum_{j=1}^{n} w_{ij} x_j$, выходная переменная x_i определяется в соответствии с (2)

$$x_i = F_i(\sum_{j=1}^{n} w_{ij} x_j - h_i), \qquad (2)$$

здесь w_{ij} - весовые коэффициенты входов; x_j - входная переменная; h_i - порог данного элемента.

Каждый такой «нейрон» имеет несколько дендритов связанных с соседними «нейронами». Сигнал x_j (входная переменная) от соответствующей точки изображения через рецептор поступает в «нейрон»,

усиливаясь положительным весовым коэффициентом центрального дендрита w_{ij}, а сигналы соответствующие соседним пикселям изображения, которые поступают через рецепторы, тормозятся отрицательными весовыми коэффициентами $-w_{ij}$ боковых дендритов (рис.1).

В зависимости от сложности распознаваемого изображения для определения его контура переменная u_i определяется в соответствии с (3) – (9),

$$u_i = \sum_{j=1}^{n} -w_{ij-1}x_{j-1}, w_{ij}x_j, -w_{ij+1}x_{j+1}, \quad (3)$$

$$u_i = \sum_{j=1}^{n} -w_{ij-n}x_{j-n}, w_{ij}x_j, -w_{ij+n}x_{j+n}, \quad (4)$$

$$u_i = \sum_{j=1}^{n} -w_{ij-(n-1)}x_{j-(n-1)}, w_{ij}x_j, -w_{ij+(n-1)}x_{j+(n-1)}, \quad (5)$$

$$u_i = \sum_{j=1}^{n} -w_{ij-(n+1)}x_{j-(n+1)}, w_{ij}x_j, -w_{ij+(n+1)}x_{j+(n+1)}, \quad (6)$$

$$u_i = \sum_{j=1}^{n} -w_{ij-(n+1)}x_{j-(n+1)}, w_{ij}x_j, -w_{ij-1}x_{j-1}, -w_{ij+(n+1)}x_{j+(n+1)}, \quad (7)$$

$$u_i = \sum_{j=1}^{n} -w_{ij-(n-1)}x_{j-(n-1)}, w_{ij}x_j, -w_{ij-n}x_{j-n}, -w_{ij+(n-2)}x_{j+(n-2)}, \quad (8)$$

$$u_i = \sum_{j=1}^{n} -w_{ij-(n-1)}x_{j-(n-1)}, w_{ij}x_j, -w_{ij-n}x_{j-n}, -w_{ij+(n-2)}x_{j+(n-2)}, -w_{ij-1} \cdot x_{j-1}, -w_{ij+1}x_{j+1}, -w_{ij+n}x_{j-n}, -w_{ij+(n+2)}x_{j+(n+2)}, -w_{ij-(n+1)}x_{j-(n+1)}, \quad (9)$$

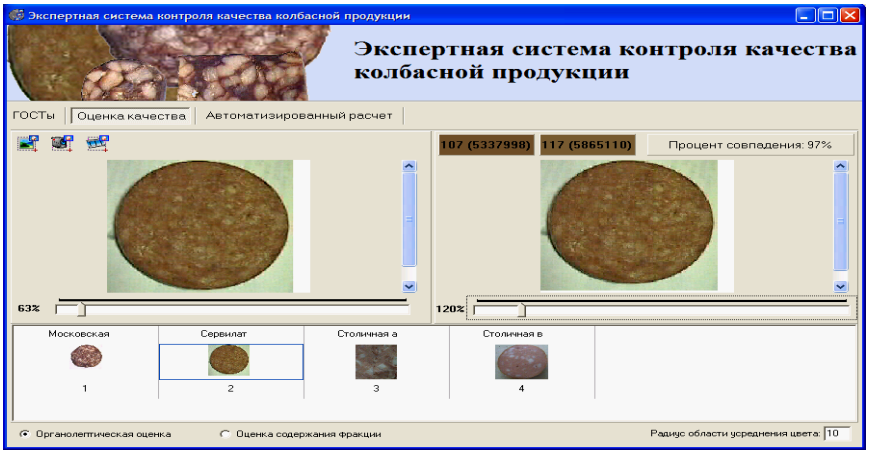


Рис.3. Исследуемая проба колбасы «Сервилат»

На выходе нейронной сети получается последовательность чисел, которая отвечают по определенному закону значениям кодов цветов входных точек изображения.

Если полученные на выходе данные изобразить в виде графика, то местам резкой смены цвета изображения соответствуют резкие перепады полученной функции. Зафиксировав эти перепады, сравниваются их размер с заданным порогом. Превышение этого размера порога свидетельствует о наличии контура изображения в данной точке.

Одновременно с расчетом контура подсчитывается усредненная цветовая гамма структуры выделяемой фракции исследуемой продукции. На рис.2 показано определение процентного содержания фракций исследуемой продукции.

Проведенные исследования показывают возможность определения и других показателей качества продукта. Так, на пробе колбасы «Сервилат» (рис.3) по структуре и цвету имеются все основания предположить о наличии в продукте добавок сои, что подтверждается и вкусовыми качествами продукции. Дальнейшие исследования позволят уточнить полученные результаты и расширить возможности системы по определению органолептических показателей качества продукции.

Заключение

В работе представлен оригинальный метод выделения контуров и структур включений различных фракций на изображениях, исследуемых проб, контролируемой продукции мясной промышленности.

Разработка и применение автоматизированных систем контроля качества мясной продукции, позволяющих исключить человеческий фактор субъективности оценок, весьма актуальны и своевременны. Проведенные исследования на тестовых программах, составляющих основу экспертной системы контроля качества мясной продукции, показывают ряд преимуществ перед существующими способами оценки качества продукции. В частности точность, безошибочность и объективность оценок, определяемых автоматически.

Литература

[Fu K., Mu J., 1981].Fu K.S., Mu J. Pattern Recognition, 1981, v.13. №1.

[Hall. 1979].Hall E. Computer Image Processing and Recognition - N.Y.: Academic Press, 1979.

[ГП «Севастопольстандартметрология», 2004]. ГП "Севастопольстандартметрология", 1999-2004, chalaya.htm.

[Дуда., Харт, 1976]. Дуда Р., Харт П. Распознавание образов и анализ сцен. Пер. с англ. под ред. В.Л. Стефанюка .– М.: Мир, 1976.

[Женкова, 2003] Женкова А. Колбаса без мяса. Но безопасная. - Киевские Ведомости.- 2003, 19 Декабря . - №282. (3087).

[Мозговая, 2005]. Мозговая О. Колбасный рай Foods & Drinks.htm. 2005.

Информация об авторе

Зайцев Вадим Геннадиевич *– предприятие «Юг энергоресурс», директор. Украина, Алушта, e-mail: misss@immsp.kiev.ua*

I.2.4. Ontologies

MULTILEVEL ONTOLOGIES FOR DOMAINS WITH COMPLICATED STRUCTURES[1]

Irene Artemieva

Abstract: *The article defines the class of domains with complicated structures, gives the definition of multilevel ontologies and determines the method for developing such ontologies.*

Keywords*: Domains with complicated structures, multilevel ontologies*

ACM Classification Keywords*: I.2.4 Knowledge Representation Formalisms and Methods, F4.1. Mathematical Logic*

Introduction

At present there are the following ontology application categories: Knowledge management systems, Controlled vocabulary, Web site or document organization and navigation support, Browsing support, Search support (semantic search), Generalization or specialization of search, Sense "disambiguation" support, Consistency checking (use of restrictions), Auto-completion, Interoperability support (information/process integration), Support validation and verification testing, Configuration support, Support for structured, comparative, and customized search [Denny, 2002], [Gavrilova, 2006], [Zagorulko et al, 2006], [Ontos Miner], [http://www.alphaworks.ibm.com/contentnr/semanticsfaqs]. The goal of research into ontologies is to create explicit, formal catalogues of knowledge that can be used by intelligent systems (see http://www.aaai.org/AITopics/html/ontol.html). The following definitions of an ontology are used:

- An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is merely a specific subject area or area of knowledge, such as petroleum, medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable. (see http://www.w3.org/TR/webont-req/).

- An ontology comprises a formal explicit description of concepts (often called classes) in a domain of discourse, properties (sometimes called slots) of each concept describing various features and attributes of the concept, and restrictions on properties (sometimes called facets). An ontology together with a set of individual instances of classes constitutes a knowledge base. In reality, there is a fine line between where the ontology ends and the knowledge base begins, and a fine line between a class and an instance. Classes are the focus of most ontologies. Classes describe concepts in the domain. (see http://www.alphaworks.ibm.com/contentnr/semanticsfaqs).

At present there are various ontology-design methodologies [Cristani & Cuel, 2005], [Denny, 2002], [Gavrilova, 2006], [Jones, et al], [Corcho et al, 2003] for the above applications. Many methodologies include defining classes in the ontology, arranging the classes in a taxonomic (subclass–superclass) hierarchy, defining slots and

[1] This paper was made according to the program № 14 of fundamental scientific research of the Presidium of the Russian Academy of Sciences, the project “Intellectual systems based on multilevel models of domains”.

describing set of values for these slots, filling in the values for slots for instances [Noy, 2001]. Different ontologies (see, for example, http://musing.deri.at/ontologies/v0.3/ and http://www.daml.org/ontologies/) designed to be used in program systems of the above classes were developed using these methodologies.

However the specified tasks are only a subset of applied tasks for the automated solutions of which intelligent systems are required, and a number of domains have characteristics that are not taken into consideration by the existing ontology-design methodologies. The aim of this article is describing the class of such domains, structure of their ontologies, and method of developing their ontologies.

Domain class definition

Domains with complicated structures have the following characteristics:

- they have sections that are described in different but resembling notion systems [Kleshchev & Artemjeva, 2005a];
- the sections have subsections that are described in different but resembling notion systems;
- any subsection can have subsections with the above characteristic.

Sections (and subsections) of domains with complicated structures are also domains with their own kinds of activity and sets of applied tasks; some applied tasks from different sections may be similar. Notions of ontologies and knowledge of different sections can be used to solve applied tasks in domains with complicated structures.

Medicine is an example of a domain with complicated structure [Kleshchev et al, 2005a]. The examples of sections of this domain are therapeutics, surgery and others. The ontology terms of each section are names of diseases studied in this section and names of signs the values of which are used to diagnose diseases. Each section has its own sets of names of diseases and signs.

Chemistry is another example of a domain with complicated structure. The examples of its sections are physical chemistry, organic chemistry and analytical chemistry. Physical chemistry deals with the physicochemical processes [Artemieva, Tsvetnikov, 2002]. These processes are described in terms of characteristics of matters and reactions that take part in the processes. Organic chemistry adds terms relating to structural properties of matters [Artemieva et al, 2005, 2006]. Analytical chemistry studies processes of influence on matters with various kinds of radiation [Artemieva, Miroshnichenko, 2005]. Chemical thermodynamics and chemical kinetics are examples of sections of physical chemistry; sections of analytical chemistry depend on analysis techniques (for example, X-ray fluorescence analysis).Another example of a domain with complicated structure is the domain of program transformations. The processes of changing programs as a result of applying different transformations are studied in this domain [Artemieva et al., 2002]. The examples of the sections are transformations of structural programs and transformations of parallel programs. The transformations are described in terms of properties of languages of these programs.

Examples of tasks solved in domains with complicated structures are diagnosing, designing, planning, etc. One needs terms required for defining characteristics of a patient (object of diagnosis) which are input data of the task, intermediate data or the result of the task and not for defining characteristics of a disease (that are stored in knowledge bases) to specify a medical diagnosis task. Terms that describe characteristics of substances used in experiments, reactions, chemical process conditions [Artemieva, Reshtanenko, 2006] and not known characteristics of substances and reactions that are traditionally stored in chemistry databases are necessary for specifying a task of planning a chemical experiment.

Some problems arise when intelligent systems for domains with complicated structures are designed. The first problem is integration of knowledge from various sections and subsections within the framework of one knowledge base. A means of such integration is ontology that must take into account that notion systems (ontologies of sections and subsections) used in different sections and subsections differ. The second problem is a way of integration of notion systems (ontologies). A means of such integration can be ontology of higher level of generality.

Defining level of generality of ontologies

Ontologies are used to verbally represent information. Verbal representation of information is a mapping of a finite set of terms into a set of possible values of terms. Verbal representation of information has level 0.

Ontology with the system of knowledge that specifies a particular set of verbal representations of information has level 1. At level 1 ontology terms has no values. Setting values to ontology terms make the verbal representation of the particular information have level 0.

Ontology without knowledge system has level 2. When different knowledge are added, different specifications of verbal representations of information are received.

Ontology in terms of which ontology of level i can be specified has level i + 1. All ontologies of levels more than 2 are metaontologies.

In every hierarchy the language of specification of ontologies has level that is 1 more than the highest level in the hierarchy (and in this hierarchy there are no ontologies of higher levels).

Let us explain the difference between level 1 and level 2. Ontology defines a set of terms used to verbally represent information, a set of possible values of each term, and relations between terms (ontological constraints). Ontology is the result of agreement among people that use the same information in their professional activity; therefore ontology is obviously the result of agreement among these people on what verbal representations in the domain have meaning. We will call a set of all verbal representations of information that have meaning conceptualization. Knowledge imposes additional constraints on a set of verbal representations and picks a subset out of conceptualization. Thus, level 2 specifies conceptualization, level 1 specifies its subset, where knowledge defines characteristics of this subset.

Ontology of the next level specifies a larger set of verbal representations of information as compared with ontology of the previous level. Transition from level i to level i-1 restricts this set and defines its subset. Transition from ontology of level i-1 to ontology of level i is considering ontology of level i-1 as verbalized information.

Let us consider domains as examples of verbalized information. Knowledge is represented verbally if it is specified as an array of pairs consisting of a term and its value. Terms included into the ontology of knowledge is used to verbally represent knowledge [Kleshchev & Artemjeva, 2006]. If verbalized information is a base of knowledge of a domain, then its representation has level 0. If this is a case, level 1 is ontology of knowledge consisting of definitions of terms with their sets of values and knowledge consistency constraints as well. Level 2 specifies sets of ontologies of knowledge.

Let us consider the example when information is the description of a state of affairs of the domain [Kleshchev & Artemjeva, 2006]. In respect of physical chemistry level 0 represents the information of a certain physicochemical process that took place at a certain period of time and under certain external conditions. To describe the process one may use the terms with the following values: “process steps”, “chemical substances at each process step”, “chemical reactions at each process step”, etc. The terms used for describing level 0 for the domain form the ontology of reality. Level 1 specifies the reality model of the given domain and describes all possible chemical processes the information about which can be represented in terms of the ontology; the representation of the information about each chemical process is not inconsistent with the ontological constraints and domain knowledge. The domain knowledge describes laws for going of chemical reactions, formation laws of substance from chemical elements, etc. Level 2 specifies the conceptualization of reality that is an idea about reality that the domain specialist has. This level defines concepts for this reality description.

Regarding X-ray fluorescence analysis, a section of analytical chemistry, level 0 represents the information about a certain physical process that took place at a certain period of time and under certain external conditions. To describe the process one may use the terms with the following values: “analytes”, “sample qualitative composition”, “percentage of an analyte in a sample” [Artemieva, Miroshnichenko, 2005]. The domain knowledge describes laws of physical processes during high-frequency electromagnetic radiation directed at a sample, values of characteristic radiation of analytes, etc.

The domain knowledge defines characteristics if its reality as sets of states of affairs that can take place in it. If the domain knowledge can be verbally represented, then the ontology of level 2 contains sets of terms for their representation. This ontology is a pair of two ontologies (reality and knowledge) and relations between them (additional ontological constraints). If the domain knowledge cannot be verbally represented (e.g. physics), then the ontology of level 2 coincides with the ontology of reality. In this case, the ontology of level 3 specifies a set of ontologies of reality.

There are domains where only part of knowledge is verbally represented. In this case, the ontology of knowledge contains terms that can represent this knowledge. This is characteristic of physical chemistry: knowledge about various properties of chemical elements (atomic weight, atomic number, etc.), physicochemical characteristics of substances (density, formula, etc.), reaction properties (e.g. catalyst) and so on is verbally represented in it. Laws of physical processes cannot be represented verbally.

Properties of multilevel ontologies for domains with complicated structures and method of their development

For the domain with complicated structure the level with the maximum number n is ontology of the domain. The level contains the terms with the help of which the ontology of the next level is determined. The transition to the next level means specifying ontology terms and ontological constraints of the next level [Artemieva, 2006]. If all the domain knowledge and ontological constraints of all the levels can be represented verbally, then the ontology of level n defines properties of all sets of terms of all ontologies of lower levels. The simplified ontology of medical diagnosis has this property [Kleshchev & Artemieva, 2005b].

The ontology of level n-1 consists of modules. Each module defines the ontology of a certain section of the domain. The ontology of level n-2 also consists of modules. Each module defines the ontology of a subsection. All the ontologies of level lower than n are modular. The domain knowledge base is also modular.

Let us describe the method for developing the multilevel ontology of the domain.

The development starts with defining verbalized information about the domain reality and terms for its verbal representation. The domain expert participate in this work. The knowledge engineer and the expert make a list of terms used for representing the reality, record meanings of terms and values, principles of representing states of affairs with their help. A set of all possible meanings is defined for each term (denotation of the term). Ontological constraints specifying constraints for a set of meanings of terms are formed (domain state of affairs consistency constraints).

A set of applied tasks of the professional activity is analyzed in order to define what information about the reality is to be verbally represented. Terms used for specifying input data and their results and terms used for representing values of intermediate data are defined. Ontological constraints specifying relations between all these terms are also defined. A set of tasks of the professional activity unambiguously defines the domain. Thus, the ontology of the reality contains terms that are used in applied tasks to specify input data and results of solutions, intermediate data, and relations between terms of the three groups. Then, the knowledge system, probably defining the reality of the domain more accurately, is to be designed.

Developing the ontology of level 2 starts with answering the question whether the domain knowledge can be represented verbally. If the answer is in the negative (i.e. the knowledge cannot be represented verbally), the knowledge system is developed in the same form as the system of ontological constraints (in terms of the ontology of the reality). If the answer is in the affirmative (i.e. the knowledge can be represented verbally), a list of terms for representing the domain knowledge is made with the help of an expert, definitions of these terms are developed, knowledge consistency constraints and interrelations between the reality and the knowledge are formulated. This list of terms for representing the domain knowledge and the set of knowledge consistency constraints form the ontology of the knowledge of this domain. If only a part of the knowledge can be represented verbally, a list of terms for representing only this part is made. The system of knowledge consists of two components: a set of assertions in terms of the ontology of the reality and mapping of a set of terms for

representing knowledge into a set of values. The ontology of the reality, the domain knowledge ontology, and interrelations between the reality and the knowledge form the ontology of level 2 for the domain.

The ontology of level 3 (and all following levels) can be developed if the ontologies of level 2 (and all the previous levels) of several sections of the domain are developed since this ontology specifies a set of ontologies of level 2. This step starts with answering whether the ontologies of level 2 can be represented verbally. If the answer is in the negative, developing stops. If the answer is in the affirmative, a list of terms for its representing is made, definitions of these terms are developed, consistency constraints for the ontology of level 2 are formulated. The task of this level and the following ones is searching for "regularities" in the ontology of the previous level, grouping terms with some similar properties into one set, formulating the term properties of these sets and relations between them.

Let us consider a fragment of the ontology of level 4 for chemistry represented by an applied logic language to exemplify the usage of a top-level ontology in the domain [Kleshchev & Artemjeva, 2005b].

1. sort Types of objects: {}N \ ∅

   Term "Types of objects" means non-empty set of names of object types of the domain.

2. (Type: Types of objects) sort Type: {}(R ∪ I ∪ N ∪ L)

   Each type of objects is a set of objects; each object can be named, represented with a number, can be logical value.

3. sort Types of object components: Types of objects → {} Types of objects

   Term "Types of object components" means the function that maps object type on non-empty set of names of object types. If Types of object components(t)=t' and t'={t'_1,t'_2} then objects from the set t'_1 or the set t'_2 that can be components of objects with type t.

4. Set of objects ≡ {(Type: Types of objects) j(Type)}

   This auxiliary term means a set of objects of all types.

5. Own properties of objects ≡ (λ(Type: Types of objects) (λ(Area of possible values: {}(Value sets ∪ {}Value corteges)) (j(Type) → Area of possible values))

   Term "Own properties of objects" means the function the argument of which is object type and the result of which is a set of functions. The argument of each function is a set of values or a set of corteges; the result is a set of functions. If Own properties of objects(t) = f1 and f1(m)=f2 then an argument of the function f2 is an object with type t and the result is an element of the set m.

6. Properties of components of given types ≡ (λ(Type1: Types of objects) (Type2: Types of object components (Type1)) (λ(Area of possible values: {}(Value sets ∪ {}Value corteges)) (Object that has type 1 → j(Type1), Object that has type 2→ Object components(Type1, Type2)(Object that has type 1)) → Area of possible values))

   Term "Properties of components of given types" means the function the arguments of which are two object types - t1 and t2, and the result is a set of functions the argument of each one is m set of values or corteges of values, and the result is a set of functions the arguments of each one is object of t1 type and object of t2 type which is a component of object of type t1, and the result is m set element.

7. sort Number of process steps: I[0,∞)

   Term "Number of process steps" means a number of steps included in a physicochemical process.

8. sort Types of process objects: {} Types of objects \ ∅

   Term "Types of process objects" means a set of object types that are considered as components of a physicochemical process.

9. Process components ≡ (λ(Type: Type of process objects) (I[1, Number of process steps] → {} {(v: Set of objects) Object type(v) = Type} \ ∅)

Term "Process components" means a function the argument of which is t object type, and the result is a set of functions the argument of each one is the number of a process step, and the result is a set of components of a process – non-empty subset of objects of type t.

10. Properties of process components ≡ (λ(Type: Types of process objects) (λ(Область возможных значений: {}(Value sets ∪ {}Value corteges)) (Step number→ I[1, Number of process steps], Process component → Process components(Type)(Step number)) → Area of possible values))

    Term "Properties of process components" means the function the argument of which is t object type, and the result is the function the argument of which is m set of values or corteges of values, and the result is the function the arguments of which are the number of a process step and a component of this step (object of type t), and the result is m set element.

Let us now consider the example of using the ontology of level 4 when defining the ontology of level 3 for X-ray fluorescence analysis [Artemieva, Miroshnichenko, 2005]. First, let us define the values of the parameters of ontology of level 4 (a set of terms of ontology of level 3).

1. Types of objects ≡ {Shells of chemical element atoms, Radiation transition of orbital electrons, Chemical elements}

   The ontology defines the objects of the given types. This set specifies types of objects that are studied in the section of the domain.

2. Types of object components ≡ (λ(Type: {Shells of chemical element atoms, Radiation transition of orbital electrons, Chemical elements}) (Type = Chemical elements ⇒ Radiation transition of orbital electrons), (Type ≠ Chemical elements ⇒ ∅}

   Energy levels and radiation transition of orbital electrons are defined for chemical elements; energy levels are defined for shells. Objects of other types do not have components.

3. Types of process objects ≡ {Chemical elements, Radiant energies}

   Types of chemical process objects are chemical elements and radiant energies.

Now let us define examples of ontological constraints that are part of ontology of level 3.

1. Shells of chemical element atoms ⊂ {}N \ ∅

   Shells of chemical element atoms is name of set. This set consists of designation for shells.

2. Chemical elements ⊂ {}N \ ∅

   Chemical elements is name of set. This set consists of designation of elements.

3. Radiation transition of orbital electrons ⊂ {}N \ ∅

   Radiation transition of orbital electrons is name of set. This set consists of designation of transitions.

Then let us define examples of terms that are part of ontology of level 3 that mean names of functions.

1. Own properties of shells ≡ Own properties of objects(Shells of chemical element atoms)

   Term "Own properties of shells" means the function the argument of which is a set of values or set of corteges of values m, and the result is the function the argument of which is shell, and the result is an element of m set.

2. Own properties of radiation transitions ≡ Own properties of objects (Radiation transition of orbital electrons)

   Term "Own properties of radiation transitions" means the function the argument of which is a set of values or set of corteges of values m, and the result is the function the argument of which is radiation transition, and the result is an element of m set.

3. Properties of radiation transition of orbital electrons of elements ≡ Properties of components of the given types (Chemical elements, Radiation transition of orbital electrons).

Term "Properties of radiation transition of orbital electrons of elements" means the function the argument of which is a set of values or set of corteges of values m, and the result is the function the arguments of which are chemical element and its radiation transition, and the result is an element of m set.

4. Properties of elements of a sample ≡ Properties of process components (Chemical elements)

   Term "Properties of elements of a sample" means the function the argument of which is a set of values or set of corteges of values m, and the result is the function the arguments of which are the number of a process step and chemical element of this step, and the result is an element of m set.

Finally let us define examples of terms that are part of ontology of level 2.

1. sort Binding energy of electrons on an energy level for an element: Properties of energy levels for an element (R(0, ∞))

   Binding energy of electrons on an energy level for an element is a function the first argument of which belongs to the set with name Chemical elements, and the second argument of which belongs to the set with name Energy level. The result of the function belongs to the set of real number.

2. sort Characteristic radiation frequency: Properties of radiation transition of orbital electrons of elements (R(0, ∞))

   Characteristic radiation frequency is a function the first argument of which belongs to the set with name Chemical elements, and the second argument of which belongs to the set with name Radiation transition of orbital electrons. The result of the function belongs to the set of real number.

3. sort Wave-length of characteristic radiation: Properties of radiation transition of orbital electrons of elements (R(0, ∞))

   Wave-length of characteristic radiation is a function the first argument of which belongs to the set with name Chemical elements, and the second argument of which belongs to the set with name Radiation transition of orbital electrons. The result of the function belongs to the set of real number.

4. sort Energy of characteristic radiation: Properties of radiation transition of orbital electrons of elements (R(0, ∞))

   Energy of characteristic radiation is a function the first argument of which belongs to the set with name Chemical elements, and the second argument of which belongs to the set with name Radiation transition of orbital electrons. The result of the function belongs to the set of real number.

Conclusion

Top-level (or upper-level) ontologies are described in many papers. Such ontologies define terms used for highly abstract notions that studied by philosophy. They are aimed at defining all meanings of these terms.

However in domains of professional activity considered in this paper it is assumed that meanings of domain ontology terms are fixed. Terms of ontology of higher level of generality defined in the article specify properties of sets of these terms and have fixed meanings themselves.

The method of development of multilevel ontologies was used to create multilevel ontology of chemistry [Artemieva et al, 2005, 2006], [Artemieva, Miroshnichenko, 2005], [Artemieva, Tsvetnikov,2002], and multilevel ontology of domain “Optimization of sequential programs” [Artemieva et al, 2002].

Properties of an intelligent system based on multilevel ontology were described in the article [Artemieva, Reshtanenko, 2006].

Bibliography

[Artemieva, 2006] Artemieva I.L. Multilevel mathematical models of domains. In Artificial Intelligence, Ukraina, 2006, vol.4: 85-94. – ISSN 1561-5359.

[Artemieva et al, 2002] Artemieva I.L., Knyazeva M.A., Kupnevich O.A. A model of a domain ontology for "Optimization of sequential computer programs". Terms for optimization process description. In 3 parts. In Scientific & Technical Information. Part 1: 2002. №12: 23-28. Part 2: 2003. №1: 22-29. Part 3: 2003, № 2: 27-34.

[Artemieva et al, 2005] Artemieva I.L., Vysotsky V.I., Reshtanenko N.V. A model for the ontology of organic chemistry. In Scientific & Technical Information, 2005, № 8: 19-27.

[Artemieva et al, 2006] Artemieva I.L., Vysotsky V.I., Reshtanenko N.V. Description of structural formula of organic compounds in the model for the ontology of organic chemistry. In Scientific & Technical Information, 2006, №2: 11-19.

[Artemieva, Miroshnichenko, 2005] Artemieva I.L., Miroshnichenko N.L. A model for the ontology of X-ray fluorescence analysis. In Informatic & Management Systems. 2005. № 2: 78-88. – ISSN 1814-2400.

[Artemieva, Reshtanenko, 2006] Artemieva I.L., Reshtanenko N.V. Specialized computer knowledge bank for organic chemistry and its development based on the ontology. In Artificial Intelligence, Ukraina, 2006, vol.4: 95-106. – ISSN 1561-5359.

[Artemieva, Tsvetnikov, 2002] Artemieva I.L., Tsvetnikov V.A. A fragment of the ontology of physical chemistry and its model // Investigated in Russia [Electronic resource]: multysubject scientific journal / Moscow Institute of Physics and Technology - Dolgoprudny: MIPT. 2002. № 5. P.454-474. - http://zhurnal.ape.relarn.ru/articles/2002/042.pdf

[Corcho et al, 2003] Corcho O., Fernandez-Lopez M., Gomez-Perez A. Methodologies, tools and languages for building ontologies. Where is their meeting point? In Data & Knowledge Engineering, 2003. № 46: 41–64.

[Cristani & Cuel, 2005] A Survey on Ontology Creation Methodologies. In Int. J. on Semantic Web & Information Systems, 2005, 1(2): 48-68.

[Denny, 2002] Denny M. Ontology Building: a Survey of Editing Tools. URL: http://www.xml.com/pub/a/2002/11/06/ontologies.html

[Jones, et al] Jones D., Bench-Capon T. and Visser P. Methodologies For Ontology Development. URL: http://www.iet.com/Projects/RKF/SME/methodologies-for-ontology-development.pdf

[Gavrilova, 2006] Gavrilova T.A. Development of applied ontologies. URL: http://raai.org/resurs/papers/kii-2006

[Kleshchev et al, 2005] Kleshchev A.S., Moskalenko F.M., Chernyakhovskaya M.Yu. A model for the ontology of medical diagnosis. In 2 parts. In Scientific & Technical Information. Part 1: 2005. №.12: 1-7. Part 2: 2006. № 2: 19-30.

[Kleshchev, Artemieva, 2005a] Kleshchev A.S., Artemieva I.L. An analysis of some relations among domain ontologies. In Int. Journal on Inf. Theories and Appl., 2005, vol 12, № 1: 85-93. – ISSN 1310-0513.

[Kleshchev, Artemieva, 2005b] Kleshchev A.S., Artemieva I.L. A mathematical apparatus for ontology simulation. In Int. Journal on Inf. Theories and Appl., 2005, vol 12, №№ 3-4 – ISSN 1310-0513.

[Kleshchev, Artemieva, 2006] Kleshchev A.S., Artemieva I.L. Domain ontologies and their mathematical models. In the Proceedings of the XII-th International Conference "Knowledge-Dialog-Solution" - KDS 2006, June 20-25, Varna, Bulgaria, Sofia: FOI-COMMERGE-2006: 107-115. – ISBN 954-16-0038-7.

[Noy, 2001] Noy N., McGuinness D. Ontology Development 101 : A Guide to Creating Your First Ontology.- Knowledge Systems Lab, Stanfo. URL: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

[Ontos Miner] Ontos Miner: Data Mining System from Text Documents in Russian. URL: http://www.avicomp.ru/rus/ontos/academic_solutions_rus.php.

[Zagorulko et al., 2006] Zagorulko Yu. A., Borovikova O.I., Kononenko I.S., Sidorova E.A. Approach to developing domain ontology for knowledge portal of computational linguistics. In Computational linguistics and intellectual technologies: Papers of International Conference "Dialogue 2006" (Bekasovo, 31 May – 4 June, 2006). – Moscow: RSUH Publishing Centre, 2006: 148-151.

Authors' Information

Irene L. Artemieva – *artemeva@iacp.dvo.ru*

Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences; 5 Radio Street, Vladivostok, Russia

MANAGEMENT OF INFORMATION ON PROGRAM FLOW ANALYSIS

Margarita Knyazeva, Dmitry Volkov

***Abstract**: The article proposes the model of management of information about program flow analysis for conducting computer experiments with program transformations. It considers the architecture and context of the flow analysis subsystem within the framework of Specialized Knowledge Bank on Program Transformations and describes the language for presenting flow analysis methods in the knowledge bank.*

***Keywords**: Knowledge bank; Ontology; Knowledge base; Ontology editor; Database editor; Flow analysis; Editor of flow analysis methods*

***ACM Classification Keywords**: I.2.5 Artificial intelligence: programming languages and software*

Introduction

The impossibility of carrying out computer experiments opportunely constitutes the main problem of program optimization science. Their goal is to determine how often transformations can be applied in real programs, what effect can be achieved, and what strategy is the best to be applied for the specified set of optimizing transformations. At present, optimizing compilers are the only means of conducting such experiments [Bacon, 1994] [GNU, 2007]. However, the period between the moment when a new transformation description is published and the moment when the realization of an optimizing compiler containing this transformation (if such a compiler is being developed) ends is so long that the results of computer experiments with this transformation appear to be out-of-date. Besides, an optimizing compiler usually contains a wide set of transformations and built-in strategy of their application so it is impossible to obtain reliable results of computer experiments related to a particular transformation (not to the whole set) or other strategy.

The absence of tools for conducting experiments results in transformations and transformation application strategies, whose characteristics are not known completely, being included in optimizing compilers. This adversely affects their making. Therefore to create a system for program transformation experiments aimed to solve the above-mentioned problems is a topical issue. Artificial intelligence methods applied in program transformations serve as a basis for this system.

Based on the results of the paper [Orlov, 2006], the paper [Kleshchev, 2005] proposes Specialized Knowledge Bank on Program Transformations (SKB_PT) as the concept of program transformation information management to solve scientific, practical and educational problems in the sphere of program transformations. This article proposes the model of management of information about knowledge-managed program flow analysis that is a tool of getting reliable information about program performance without its execution in the program transformation system in SKB_PT. The multipurpose computer knowledge bank is used as the general concept within the framework of which the program transformation system is realized with the knowledge-managed program flow analysis [Orlov, 2006] (http://www.iacp.dvo.ru/es/mpkbank).

The paper has been financially supported by the Far Eastern Branch of the Russian Academy of Sciences, initiative-based research project “Internet system for controlling information about program transformations”.

Concept of knowledge-managed program flow analysis

The extraction of certain semantic characteristics of a program takes place during flow analysis that is traditionally divided into control flow analysis and data flow analysis [Kasyanov, 1988] [Voevodin, 2002].

The main task of control flow analysis is to present and structure sets of program executions, to find characteristics of statements and branches in these executions, to choose an order of program statements

processing. During data flow analysis each program is executed in parallel over all values from a symbolic and very simplified version of its real data area.

Let us consider the architecture of the subsystem of the knowledge-managed flow analysis within the framework of the program transformation system (fig. 1).

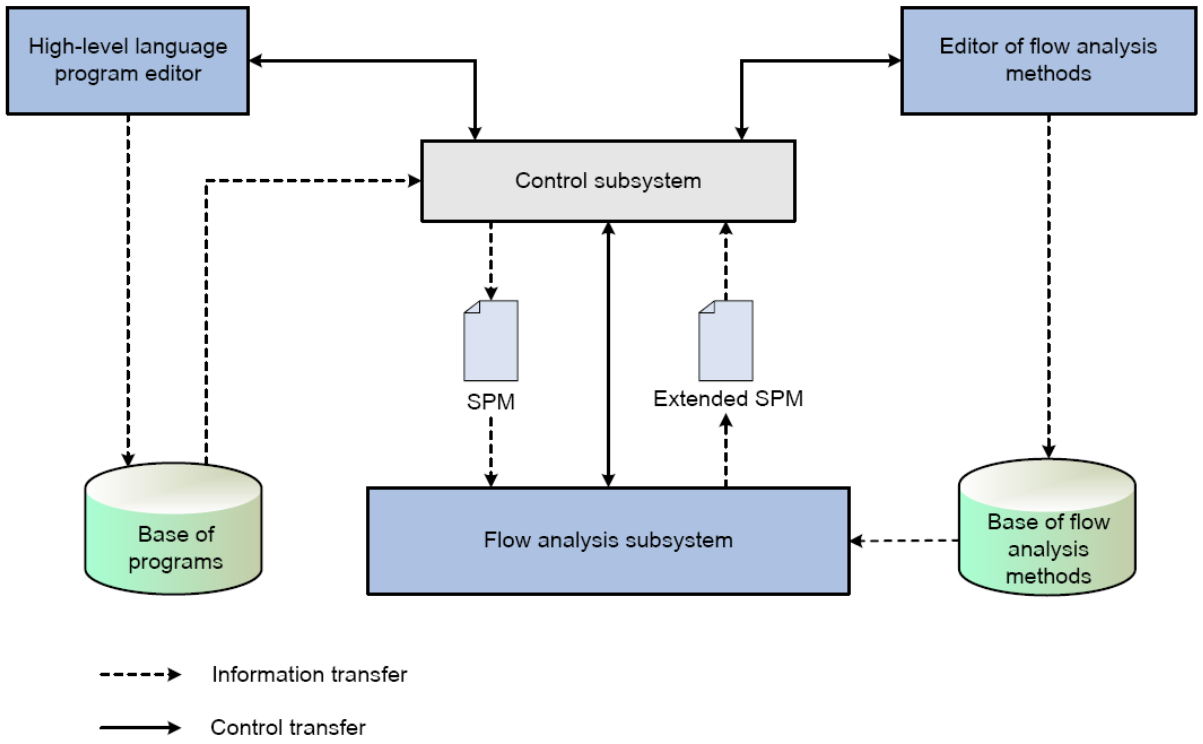


Fig. 1. Architecture of subsystem of knowledge-managed flow analysis

Structural program model (SPM), flow analysis methods and task to do a program flow analysis are input data of a knowledge-managed flow analysis subsystem in the system of program transformation. SPM extended with the terms of the program flow analysis is formed at the output of the subsystem.

Structural program model defined in [Knyazeva, 2005a] is a single internal presentation at which the program flow analysis takes place. It is presented as a graph. Extended SPM is control and information graphs of the program [Knyazeva, 2005b]. To extend SPM is to add special control arcs to the program presentation and enter new program fragments which result from the program flow analysis into SPM. Extended SPM is the basis for program transformations. Functions and relationships assigning some program characteristics are defined on a set of fragments and identifiers of the program. Functions that have one argument are called attributes.

In order to apply new flow analysis methods in experiments, the flow analysis subsystem gives the user the opportunity to exploit a specialized language, that describes flow analysis methods (FAM), to assign methods of program flow analysis.

The task to do a flow analysis is a description of the knowledge out of the whole volume of the knowledge about flow analysis methods that are to be applied in this situation.

Source data are entered into the corresponding databases by means of High-level language program editor and Editor of flow analysis methods. Control subsystem provides the interaction between the flow analysis subsystem, program transformation system and data sources.

The base of programs contains high-level language programs in terms of language ontologies.

The base of flow analysis methods contains flow analysis methods in the language of flow analysis methods.

Language of flow analysis methods

Main forms of notation of flow analysis methods described in relevant works were analyzed when the language being developed. It contains variables that may take on references to various elements of the program model as values; basic constructions of algorithmic programming languages (such as loop, selection, assignment); operations with sets as variable sets and identifier sets are operated on while the information about the program is being accumulated; tree walk operations and operations with tree structures.

The syntax of the language of flow analysis methods is described in extended BNF notation:

<Flow analysis method>::= " Flow_analysis_method" "(" <Method name> ")" <Variable declaration block>
<Sequence of constructions>
< Method name >::=<String>
<String>::=<Letter> | <String> < Letter > | <String> <Digit>
< Letter >::=A | ... | Z | a | ... | z| - | _ |
< Digit >::=0 | 1 … | 9
<Sequence of constructions>::= "{" [<Construction>] "}"
< Variable declaration block >::= [<Variable declaration >]
< Variable declaration >::= <Variable type> ":" (<Variable-fragment> | <Variable-attribute> | <Variable-arc> | <Variable-relation> | <Variable> [","]) ";"
<Variable type>::="Variable-fragment" | "Variable-attribute" | "Variable-arc" | "Variable-relation" | "Integer" | "Real"
<Construction>::=<Formula> | < Walk> | <Selection> | <Loop> | <Assignment> | <Program modification>
<Formula>::=<Formula with fragment> | <Formula with set> | <Logical formula>
< Walk>::=<Program tree walk> | <Expression tree walk>
<Selection>::="If" "(" <Logical formula> ")"
"Then" <Sequence of constructions> ["Else" <Sequence of constructions>]
<Loop>::="While" <Condition> <Sequence of constructions>
<Assignment>::= <Left part of assignment> "=" <Right part of assignment>
<Program modification>::= <Fragment creation> | <Attribute creation> | <Attribute change> | <Arc creation> | <Relation creation> | <Variable creation>
<Formula with fragment>::= <Arc fragment> | <Fragment attribute> | <To get class> | <To get expression variable> | <First arc fragment of sequence> | <Next arc fragment of sequence>
<Arc fragment>::= " Arc_fragment" "("<Variable-fragment>, <Arc name>, <Variable-fragment> ")"
<Fragment attribute>::= "Fragment_attribute" "(" <Variable-fragment>,
<Attribute name>, <Variable-attribute> ")"
<To get class>::= "To_get_class" "(" <Variable-fragment>, <Fragment class>")"
<To get expression variable>::= "To_get_expression_variable" "(" <Variable-fragment>, <Variable> ")"
<First arc fragment of sequence>::= " First_arc_fragment_of_sequence" "(" <Variable-fragment>, <Variable-fragment> ")"
<Next arc fragment of sequence>::= "Next_arc_fragment_of_sequence" "(" <Variable-fragment>, <Variable-fragment>, <Variable-fragment> ")"
<Formula with set>::= <Intersection of sets> | <Union of sets> | <Equality of sets>
<Intersection of sets>::= "Intersection_of_sets" "(" <Variable-set> <Variable-set> < Variable-set> ")"
<Union of sets>::=" Union_of_sets" "(" <Variable-set> <Variable-set> <Variable-set> ")"
<Equality of sets>::= " Equality_of_sets" "(" <Argument-set> <Variable-set> <Boolean-set> ")"
<Logical formula>::=<Term of logical formula>
<Compound logical formula>::= <Term of logical formula>
<Logical operator> <Term of logical formula>

<Term of logical formula>::=<Compound logical formula> | <Boolean set> | < Equality of sets> | <Fragment class> | <Arc name> | <Variable-fragment> | <Variable-attribute> | <Variable-arc> | <Variable-relation> | <Attribute name> | <Relation name> | <Variable>
<Logical operator>::= “>” | “<” | “>=” | “<=” | “<>” | “==” | “AND” | “OR” | “NOT”
<Program tree walk>::=“Program_tree_walk” “(“ <Variable-fragment>, <Variable-fragment>, <Logical formula> “)” <Sequence of constructions>
<Expression tree walk>::=“Expression_tree_walk” “(“<Variable-fragment>, <Variable-fragment> “)” <Sequence of constructions>
<Program modification>::=<Fragment creation> | <Attribute creation> | <Arc creation> | <Relation creation> | <Variable creation>
<Fragment creation>::=“To_create_fragment” “(“ <Variable-fragment>,<Fragment class> “;”
<Variable-fragment> “)”
<Attribute creation>::= “To_create_attribute” “(“ <Variable-fragment>, <Attribute name>, <Variable-attribute> “)”
<Arc creation>::=“To_create_arc” “(“ <Variable-fragment>, <Variable-fragment>, <Arc name>, <Varaiable-arc> “)”
<Relation creation>::= “To_create_relation” “(“ <Variable-fragment>, <Variable-fragment>[, <Variable-fragment><Variable-relation> “)”
<Value>::=<Integer> <Real> <Boolean set>
<Integer>::=(<Digit>)
<Real>::=(<Digit>)[,(<Digit>)]
<Assignment>::=<Left part of assignment> = <Right part of assignment>
<Left part of assignment>::=<Variable-fragment> | <Variable-attribute> | <Variable-arc> | <Variable-relation> | <Variable>
<Right part of assignment>::=<Variable-fragment> | <Variable-attribute> | <Variable-arc> | <Variable-relation> | <Variable> | <Value> | <Arithmetic expression>
<Arithmetic expression>::=<Term of arithmetic expression> <Arithmetic operator> <Term of arithmetic expression>
<Term of arithmetic expression>::=<Arithmetic expression> <Variable> <Bracketed arithmetic expression> <Variable value>
<Arithmetic operator>::= “+” | “- ” | “* ” | “/ ” | “^”
<Fragment class>::=“Variable_declaration” | “Function_declaration” | “Parameter_declaration” | “Variables_declaration” | “Functions_declaration” | “Parameters_declaration” | “Assignment” | “Input” | “Output” | “Program_block” | “Conditional_statement” | “Loop_with_step” | “Loop_with_precondition” | “Loop_with_postcondition” | “Procedure_call” | “Dynamic_variable_elimination” | “Expression” | “Sequence_of_statements”
<Attribute name>::=“Reverse_Polish_notation” | “Result_array” | “Pointer” | “Function_recursive” | “Side_effect” | “Reference_to_memory_space” | “Nesting_level” | “Priority” | “Type” | “Reference_parameters” | “Value_parameters” | “Actual_reference_parameters” | “Changeable_actual_reference_parameters” | “Actual_value_parameters” | “Argument_set” | “Result_set” | “Obligatory_result_set” | “Function_declaration_statement” | “Contiguous_sequence_of_fragments” | “Classes_of_fragments_of_sequences” | “Quantity_of_fragments” | “Result_identifier” | “Pseudovariable” | “Design_of_new_types” | “Acceptable_left_expression”
<Arc name>::=“If” | “Then” | “Else” | “Condition_of_loop” | “For” | “Until” | “Step” | “Statement_body” | “Parameter_block” | “Local_parameter_block” | “Embedded_function_block” | “Right_expression” | “Left_expression” | “First_element_of_sequence” | “Lasy_element_of_sequence” | “Arc_statement_sequence” | “Matches_fragments” | “Next_fragment” | “Parameter_list”
<Relation name>::=“Immediate_precedence” | “Precedence” | “Similarity” | “To_be_part” | “To_be_submodel” | “Precedence_of_submodels” | “Joint_sequence” | “Intermediate_sequence” | “Preceding_sequence” | “Next_sequence”

<Boolean-set>::="true" | "false"
<Variable>::=<String>
<Variable-set>::=<String>
<Variable-fragment>::=<String>
<Variable-attribute>::=<String>
<Variable-arc>::=<String>
<Variable-relation>::=<String>

Example of presenting flow analysis method in FAM language

Context conditions for transformations are described either in terms of a program model or in terms derived from them. The model is to semantically ensure formulating of the context of the current transformation. The justification for the transformation lies in proving the theorem that context conditions for transformations are sufficient conditions for functional equivalency of transformed and source program models [Pottosin, 1980].

The enclosure of any optimizing transformation into the compiler assumes simultaneous forming of the transformation and context condition; provided the condition is met, the given transformation is applied to the program.

This can be exemplified by argument set that is a set of variables the values of which may affect a statement performance in the program. The information about argument sets of statements is made use of in optimizing transformations "unused variable elimination", "loop invariant statement removal" and others [Bacon, 1994]. The correct selection of an optimization area in a source program and the transformation efficiency on the whole depend on the flow analysis quality.

Method of copying argument set of each program statement into result set of program in FAM language:

```
Flow_analysis_method(Copying_of_sets)
Type-fragment: Current_fragment;
Type-attribute: Temporary_attribute;
{
Program_tree_walk(Function_Main; Current_fragment; Fragment_class(Current_fragment) ==
Assignment){
Fragment_attribute(Current_fragment; Argument_set; Temporary_attribute);
Attribute_creation(Current_fragment; Result_set; Temporary_attribute);
}
}
```

The first string is as follows:

```
Flow_analysis_method(Copying_of_sets)
```

The first sentence in the FAM language starts with the key word Flow_analysis_method that is followed by the flow analysis method name in parenthesis.

The section declaring variables follows:

```
Type-fragment: Current_fragment;
Type-attribute: Temporary_attribute;
```

In this example there are two variables described: the first one has Type-fragment type and is called Current_fragment, the second one has Type-attribute type and is called Temporary_attribute. Type-fragment variable type means that this variable may take on a reference to a fragment of a particular program on SPM or an object in the memory that reflects all characteristics of SPM fragment. The declarations of variables of different types are separated with semicolons. If there are declarations of several variables of one type, they can be separated with commas.

The method body immediately follows the variable declarations and consists of a sequence of constructions:

```
{
Program_walk_tree(…)
{
Fragment_attribute(…);
Attribute_creation(…);
}
}
```

The method body is in braces. The inside constructions are separated with semicolons. In this example, the method body consists of one Program_walk_tree construction which consists of two constructions: Fragment_attribute and Attribute_creation.

Program_walk_tree construction is a function with three arguments that follows the key word. They are in parentheses and separated with a semicolon and the body in braces:

```
Program_walk_tree (Function_Main; Current_fragment; Fragment_class(Current_fragment) ==
Assignment)
{
…}
```

This function realizes SPM fragments tree walk. The first argument specifies SPM fragment which is the root the subtree that is to be walked. The second argument is a variable that takes on the fragment value at the next walk step. The third argument is a logical formula whose verity ensures the execution of the body constructions sequence. In this example, the first argument is Function_Main constant the value of which is a reference to SPM root fragment. The second argument is Current_fragment variable that takes on the next fragment value at each walk step. The third argument is a logical formula. It takes on the verity value if SPM fragment, Current_fragment refers to, has Assignment class.

The construction Fragment_attribute is a function with three arguments:

Fragment_attribute(Current_fragment, Argument set, Temporary_attribute);

The first argument is SPM fragment Current_fragment variable refers to. The second argument is the name of SPM argument that is necessary to get. The third argument is a Type-attribute variable which is the result of the function and takes on the value of the reference to the specified attribute of the current fragment.

Fragment_creation construction is a function with three arguments:

Fragment_creation(Current_fragment, Result_set, Temporary_attribute);

This function creates the attribute with the specified name and value for the specified fragment. The first argument is SPM fragment Current_fragment variable refers to. The second argument is the name of SPM attribute that is necessary to create; in this case it is Result_set. The third argument is a Type-attribute variable whose value is to be copied for a newly-created attribute.

Conclusion and Acknowledgements

This paper presents the knowledge-managed flow analysis concept. It provides examples how various flow analysis methods can be defined by means of the described language. At present, based on the knowledge-managed flow analysis concept, the flow analysis subsystem within the framework of the program transformation system in SKB_PT is developed.

Bibliography

[Bacon, 1994] Bacon D.F., Graham S.L., Sharp O.J. Compiler transformations for high-performance computing //ACM Computing Surveys 1994 V.26 № 4. PP.345-420/

[GNU, 2007] GNU Compilers Collection 3.3.2. http://gcc.gnu.org/onlinedocs/gcc-3.3.2/gcc/

[Kasyanov, 1988] Kasyanov V. N. Optimizing transformations of the programs. Moskow: Nauka, 1988. 336 p. (In Russian).

[Kleshchev, 2005] Kleshchev A.S., Knyazeva M.A. Controlling Information on Program Transformations: I. Analysis of Problems and Ways of Their Solution with methods of Artificial Intelligence. Journal of Computer and Systems Sciences International, Vol.44, No5, 2005, pp. 784-792.

[Knyazeva, 2005a] Knyazeva M.A., Kupnevich O.A. Domain ontology model for the domain "Sequential program optimization". Defining the language of structural program model. In The Scientific and Technical Information, Ser. 2.-2005.-№ 2.-P. 17-21. (In Russian).

[Knyazeva, 2005b] Knyazeva M.A., Kupnevich O.A. Domain ontology model for the domain "Sequential program optimization". Defining the extension of the language of structural program model with flow analysis terms. In The Scientific and Technical Information, Ser. 2.-2005.-№ 4. (In Russian).

[Orlov, 2006] Orlov V.A., Kleshchev A.S. Computer banks of knowledge. Multi-purpose bank of knowledge. In The Information Technologies. 2006. №2. P.2-8. (In Russian).

[Pottosin, 1980] Pottosin I.V., Yugrinova O.V. Justification for purging transformations for loops. In The Programming 1980. - №5. - P.3 - 16. (In Russian).

[Voevodin, 2002] Voevodin V.V., Voevodin Vl.V. Parallel computing. Saint Petersburg: BHV-Pereburg, 2002. 608 p. (In Russian).

Authors' Information:

Margarita A. Knyazeva, Dmitry A. Volkov *- Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences, 5 Radio Street, Vladivostok, Russia mak@nt.pin.dvgu.ru, vd2000@mail.ru*

AUTOMATIC GENERATION OF CONTEXT-SENSITIVE HELP USING A USER INTERFACE PROJECT

Valeriya Gribova

Abstract. *The article presents a new approach to automatic generation of help in software. Help generation is realized in the framework of the tool for development and automatic generation of user interfaces based on ontologies. The principal features of the approach are: support for context-sensitive help, automatic generation of help using an interface project and an expandable system of help generation.*

Keywords*: Ontology, interface project, user interface development*

ACM Classification Keywords*: I.2.2 Artificial intelligence: automatic programming*

Introduction

One of the basic quality criteria of any software is learnability (reducing learning time). A principal characteristic of learnability is intuitive understandability of a user interface, however, a help system in software is also required. Complexity and functionality of software are increasing every year. As a result, the number of users who know all features of an application program (according to statistics, users are familiar with about 10% of application program functions) is decreasing. Therefore availability of a help system for users is important feature software.

Development of a help system is a costly and time-consuming task. Most of software has some context-free help realized as static guidance or a tutor. Static guidance can be realized in the form of an on-line help system. Nevertheless, searching information in large help systems is difficult for users, and cost of maintenance is very high (when an application program is modified, the help system must be modified as well). Tutors also have shortcomings because they teach users only some aspects of using an application program and have high cost of maintenance in the life cycle of an application program. Context-sensitive help realized in some model-based interface development environments [1,2] has a number of advantages over context-free help. The most important of them are automatic generation of a help system and using a current status execution of an application program when an answer in the help system is being generated. Nevertheless, the grave disadvantage of this system is that it cannot be expanded.

In this article an approach to automatic generation of expandable context-sensitive help is proposed. The help generator is a component of the tool for development and automated generation of user interfaces based on ontologies [3]. To provide help the help generator uses a task project. It is a component of the interface project to be used to generate the executive code of the interface. The task project is a tree. To expand the number of context-sensitive help types a script language has been developed. Recently the context-sensitive help generator has been implemented and introduced into the tool for development and automated generation of user interfaces based on ontologies.

Tools for help generation

There are next types of help in software, namely, context-free and context-sensitive. Traditional help systems are context-free and realized either in the form of a static guidance system or of a tutor.

A static guidance system provides help that is defined as a canned text at the development stage. To learn some aspects of software features users have to read a part or parts of the guidance. There are several kinds of this help: books, instruction manuals and on-line static help. On-line help is developed by authoring tools in various formats (HTML help, HTML-based Help, JavaHelp, Oracle Help, Adobe Portable Document Format – PDF, Macromedia Flash, WinHelp, AP Help, and others).

Compared with instruction manuals, on-line static help is more convenient. However, it is difficult to find required information for large on-line static help so developers make a particular section called ”help for help”. In some cases it is useful to know a special query language. The help system and the application program are not interrelated.

Tutors simulate some application program behavior; so make the process of learning easy for the user. However, they teach users only the main functions of an application program. To find out other functions of the application program, users have to use instruction manuals and on-line static help. Development and modification of tutors is expensive and time-consuming, because all the alterations in the application program must be represented in the tutor.

Context-sensitive help is a kind of on-line help. There are two types of the context-sensitive help: conceptual and procedural. The conceptual help describes a framework of an application program, knowledge required to interact with it and the meaning of interface elements. The procedural help describes functions and operations of an application program. Unlike the conceptual help, it concentrates on user’s tasks.

Implementation of the context-sensitive help is an important but very expensive task because it addresses to internal data structures. The most difficult objective is to provide conformity between the help system and the application program during the life cycle.

Some of the model-based user interface development environments (MB-IDE) have facilities for generation of context-sensitive help using an interface model. The principal method of context-sensitive help generation is based on transition networks. The transition network consists of a set of vertexes and arcs. Vertexes describe a set of valid states. Arcs show allowable user’s actions. A help generator scans the transition network to form the context-sensitive help. A number of MB-IDE use a task model for help generation. The task model represents the

tasks that users need to perform with the application program. It describes tasks by hierarchically decomposing each task into sub-tasks (steps) until leaf tasks become operations supported by the application program. Various specification languages are used for performing task models, for example, LOTOS, CCS [4,5].

Usually help systems can give answers to a set of questions which are task-related (*):

- Why this task is not available?
- How to realize this task?
- How to activate this task?
- What tasks can I perform now?

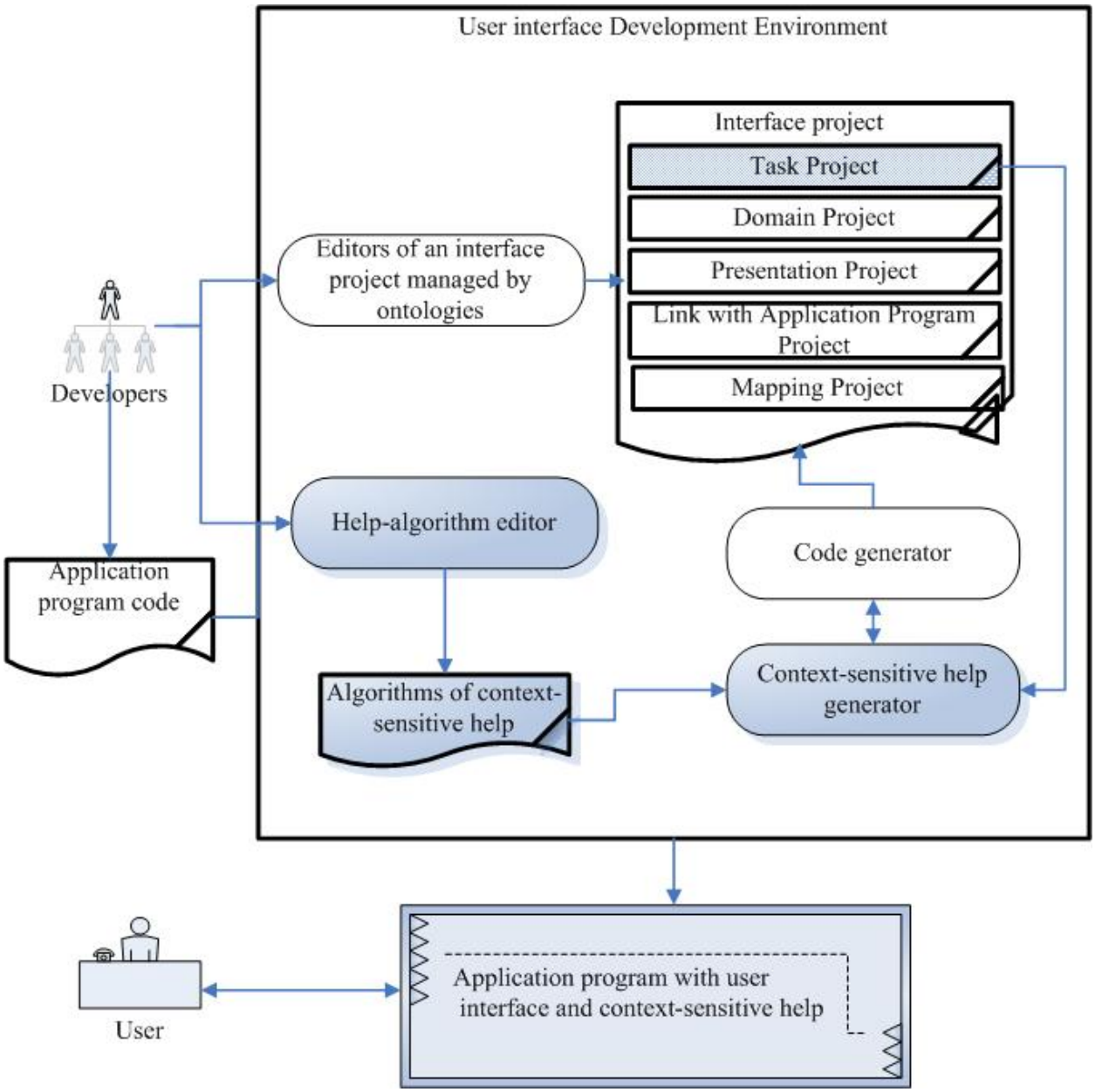


Fig. 1 The basic architecture of the expanding system of context-sensitive help-generation

The basic advantages of context-sensitive help are coupling with the interface model (the task model is a component of an interface model) and automatic generation of help which makes the cost of development and maintenance of context-sensitive lower compared with other types of help.

Nevertheless, help systems support only a fixed number of questions. To add a new question to a system it is necessary to develop a new version of the help system. Tasks do not have an executing status so the quality of the help is lowered.

An approach to help generation

The state of the art in help generation enables us to set following some principle:

- Help should be context-sensitive;
- Help should be automatically generated at the time of user interface generation;
- Help should be invoked in any period of user's interaction with a user interface;
- Help should give answers to the set of questions marked * (see above);
- A help system should be expandable.

Context-sensitive help is realized in the framework of the tool for development and automatic generation of user interfaces based on ontologies (Fig. 1).The main idea of an ontology-based approach [3] to user interface development and generation is to form an interface project using ontology models which describe features of every component of the project and then, based on this project, generate a code of the user interface. The components of the interface project are:

- A domain project,
- A task project,
- A presentation project,

A project of link a user interface with the application program (An application program project),

- A dialog scenario project,
- A mapping project.

Every component of the interface project is developed by a structural or graphical editor managed by an ontology model.

The domain project determines domain terms, their properties and relations between them. These terms describe output and input data of the application program and information on the intellectual support of the user.

The task project determines the tasks users can implement using the application program.

The presentation project determines a visual component of the interface and provides support for various types of the dialog.

The an application program project determines variables, types of their values shared by the interface and the application program, protocols for communication between the application program and the interface, addresses of servers and methods of messages transfer.

The dialog scenario project determines abstract terms used to describe the response to events (sets of actions executed when an event occurs, sources of events, modes of transfer between windows, methods of the window sample selection, and so on).

The mapping project determines relations between components of the interface project.

A context –sensitive help generator uses the task project of the interface project and a current task (the name of an executed task). The task project is a tree. The root of the tree is marked by the name of an abstract task which represents a set of application program tasks. The task project can be reused to design other interfaces if their application programs have similar features. Nonterminal vertexes of the tree are marked by names of abstract

tasks. Terminal vertexes are marked by names of elemental tasks. The elemental tasks are tasks that cannot be divided into sub-tasks. Arcs of the tree do not have any marks; they link vertexes indicating the tasks hierarchy.

Every set Y can be divided into four types of sub-sets. The set Y={Y1, Y2,…,YN} is formed from marks of vertexes which are direct descendants of a vertex marked by X. Every sub-set of the Y set establishes relations among tasks. These relations are: choice, interleaving, synchronization and deactivation. These relations are taken from notations developed for specifying concurrent systems (LOTOS) [4].

- The sub-set called “choice” means that every task from the sub-set can be implemented; nevertheless, users can not start implementing any task until the previous task, i.e. the task which has begun implementation has been completed.
- The sub-set called “interleaving” means that every task from the sub-set can be implemented; users can start implementing any task of the sub-set while the previous task from this sub-set is been executed.
- The sub-set called “synchronization” means that only successful implementation of the sub-set task allows the user to implement other tasks of the sub-set.
- The sub-set called “deactivation” means that as soon as a task from the sub-set has been completed, implementation of the other tasks of the sub-set is broken.

A process of the interface project design according to the ontology-based approach, in particular, requires that the developer of a user interface in the mapping project determine links between every task from the task project and interface elements from the presentation project. By handling an interface element, the user initiates implementation of a task linked to this interface element. As soon as an event of the interface element occurs, implementation the task begins. It means execution of an action chain defined in the scenario dialog project. For example, a task called “activation of the expert system” is defined in the task project. This task links to a push-button (an interface element) called, “activation of the expert system”, in the mapping project. When an event called “keystroke” appears in the interface, i.e. it means that the user presses the push-button called by the name of this task, “activation of the expert system”, the following chain of actions defined in the scenario dialog project begins to be executed: “to save data”, “to pass data to the application program”, “to represent new dialog window”, etc.

To realize context-sensitive help, every chain of actions in the scenario dialog project must be extended by a system function. It passes the name of the executing task to the help generator. Inclusion of this function in the scenario dialog project automatically produces a sub-system of the help generator. Then, while interacting with an application program, the user invokes context-sensitive help. The help generator based on the user’s query chooses an appropriate algorithm for help generation. Input data for this algorithm are: the name of the executing task and the task tree (the task project). To generate context-sensitive help the developer adds a set of interface elements used to call context-sensitive help to the interface project.

The principal requirement to the system of help generation is its expandability. To realize this requirement a script language for describing algorithms of help generation is proposed. This language consists of imperative constructions and a set of system functions which allow a new algorithm to be described. Using a structural editor the developer can add a new algorithm (a new kind of help generation). The added algorithm is transmitted to the XML-file and included in the algorithm base. Recently the main kinds of help generation mentioned in the requirements (see above) have been developed.

Conclusion

In this article an approach to automatic generation of context-sensitive help is proposed. The basic idea of the approach is to add an expanding system of help-generation to the tool for user interface development based on ontologies. The main task of the system is to form answers to user’s queries using a name of the executed task and the task project. To date a prototype of the system has been developed at the Intellectual Systems Department of the Institute for Automation and Control Processes, the Far Eastern Branch, the Russian Academy of Sciences.

Acknowledgements

The research was supported by the Far Eastern Branch of Russian Academy of Science, the grants «An expandable system for quality monitoring».

Bibliography

[1] Moriyón, R., Szekely, P., Neches, R.: Automatic Generation of Help from Interface Design Models. In C. Plaisant (ed.): Proceedings of CHI'94. New York: ACM Press 1994 (pp. 225-231).

[2] Palanque, P., Bastide, R.: Contextual Help for Free with Formal Dialogue Design. In Alty J.L., Diaper D., Guest S. (eds.): Proceedings of HCI'93. Cambridge: Cambridge University Press 1993.

[3] Gribova V., Kleshchev A. From an ontology-oriented approach conception to user interface development International //Journal Information theories & applications. 2003. vol. 10, num.1, p. 87-94.

[4] Paternó, F., Faconti, G.: On the Use of LOTOS to Describe Graphical Interaction. In Monk A., Diaper D., Harrison M.D. (eds.): Proceedings of HCI'92. Cambridge: Cambridge University Press 1992 (pp. 155-174).

[5] Sukaviriya, P., Foley, J.D.: Coupling a UI Framework with Automatic Generation of Context-Sensitive Animated Help. In Proceedings of UIST'90. New York: ACM Press 1990 (pp. 152-166).

Author's Information

Gribova Valeriya – *Ph.D. Senior Researcher of the Intellectual System Department, Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of the Sciences: Vladivostok, +7 (4323) 314001 e-mail: gribova@iacp.dvo.ru, http://www.iacp.dvo.ru/is.*

IN SEARCH OF A VISION: ONTOLOGICAL VIEW ON USER MODELLING CONFERENCES' SCOPE

Tatiana Gavrilova, Seppo Puuronen

Abstract: *The paper continues the research of user modeling (UM) field on visionary level. It proposes possible views on meta-ontology of the user modelling field based on the results of a workshop and analysis of a call for papers. Ontology is meant to structure the state-of-the-art in the field and serve as a central reference point and as a basis to index systems, papers and learning media. The domain has the bias on ubiquitous UM. The paper presents more questions than answers. It is a bit provocative as it is intended stir a debate within UM community of researchers.*

Keywords*: Ontology Design, Knowledge, Educational Ontology, C Programming, Ontology Visualization.*

Introduction

Starting research in the field of user modelling (UM) is a challenge. The area of the user modelling research is a rather young field and as all human-computer interaction study possesses a great deal of polysemanticism and heterogeneity. The terminology is full of contradictions, multiple meanings and not standardized yet. A lot of terms have several synonyms (e.g. decentralized/distributed, personalized/individualized, etc.) and some terms are rather fuzzy (e.g. generic user model). Every researcher creates his/her own vocabulary and ontology of the domain and tries to share it with others on the conferences and workshops.

Conferences help us to re-structure, to augment or to share our vision (if we have any). Young researchers mainly undervalue the broader overview and their works suffer from shallow bias. This paper presents a

framework that may be helpful for students and young researchers to have a broader view on the field in their endeavours in the field. In addition, it details an approach to promoting integrity for the research community. Providing a mind map of two workshops in ubiquitous and decentralized user modelling (Workshop “Ubiquitus User modeling UBIQUM” within European Conference on Artificial Intelligence ECAI 2006, Riva del Garda, Italy, and International Workshop on Ubiquitous and Decentralized User Modeling UbiDeUM'2007 within the 11th International Conference on User Modeling UM 2007, Corfu, Greece) this paper may be essential for anyone concerned the scientific study in the field.

There exist a lot of approaches to UM but a common schema that would attempt to classify them all has not been proposed yet. Such lack of structure makes attempts to conduct novel research or implement known approaches in the area of UM quite a demanding task.

This is why we dared to propose a classification, an ontology of UM field [3, 5] that may lead towards a central reference point of UM field, in a similar way as ACM computing classification system acts as a general reference point (ACM CCS) [1]. The development of UM Meta Ontology (UMMO) was a part of wider research aimed at development of user model centered learning portal. UMMO is an attempt to externalize the current approaches, techniques, and tools [2,7,8].

In this paper we elaborate that step by proposing an ontological view on the papers and topics of above mentioned workshops. Such ontology helps to present in a visual structured form the current state of the art and may serve as a teaching tool or the basis for comparing papers and pieces of research in the specific area of UM.

Mind-mapping as Ontology Design

A mind map [4] is a diagram used to represent words, ideas, tasks or other items linked to and arranged radially around a central key word or idea. It is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, and decision making [9].

Well-structured mind map can work as a visual draft of domain ontology. The process of ontology development may be guided by a recipe [6] that can be shortly summarized down to the following steps:

- glossary development,
- concept laddering, and
- visual mapping (balance, harmony, clarity).

Ontological view on UBIQUM 2006

We tried to develop ontological structure of UBIQUM workshop. The programme included the following contributions:

1. "Efficient Text Summarization for Web Browsing On Mobile Devices" by Gaпl Dias and Bruno Conde
2. "Creating Ontology for User Modelling Research" by Tatiana Gavrilova, Peter Brusilovsky, Michael Yudelson and Seppo Puuronen
3. "Exchanging Personal Information" by Christian Wartena, Peter Fennema and Rogier Brussee
4. "Adaptation of Ubiquitous User-Models" Andreas Lorenz, Andreas Zimmermann
5. "User Modelling in a Distributed Multi-Modal Application" by Andreas Zimmermann, Andreas Lorenz
6. "Web Services and Semantic Web for Adaptive Systems" by Francesca Carmagnola, Federica Cena, Cristina Gena, Ilaria Torre
7. "Case-Based to Content-Based User Model Mediation" by Shlomo Berkovsky, Ariel Gorfinkel, Tsvi Kuflik and Larry Manevitz
8. "Ambient Audio Notification with Personalized Music" by Ralf Jung and Dominik Heckmann
9. "Ontology-Based User Modeling for Pedestrian Navigation Systems" by Panayotis Kikiras, Vassileios Tsetsos and Stathes Hadjiefthymiades

10. "Exploiting the Link Between Personal, Augmented Memories and Ubiquitous User Modeling" by Alexander Kroner and Dominik Heckmann and Michael Schneider
11. "An Agent-Based Approach Supporting Personal Ubiquitous Interaction" by Francesca Muci, Pawel Drozda, Giovanni Cozzolongo
12. "Towards Situated Public Displays as Multicast Systems" by Hans Jorg Muller and Antonio Kruger

displays one of many possible visions of the UBIQUM ("Ubiquitous User modeling UBIQUM" within European Conference on Artificial Intelligence ECAI 2006, Riva del Garda, Italy) workshop.

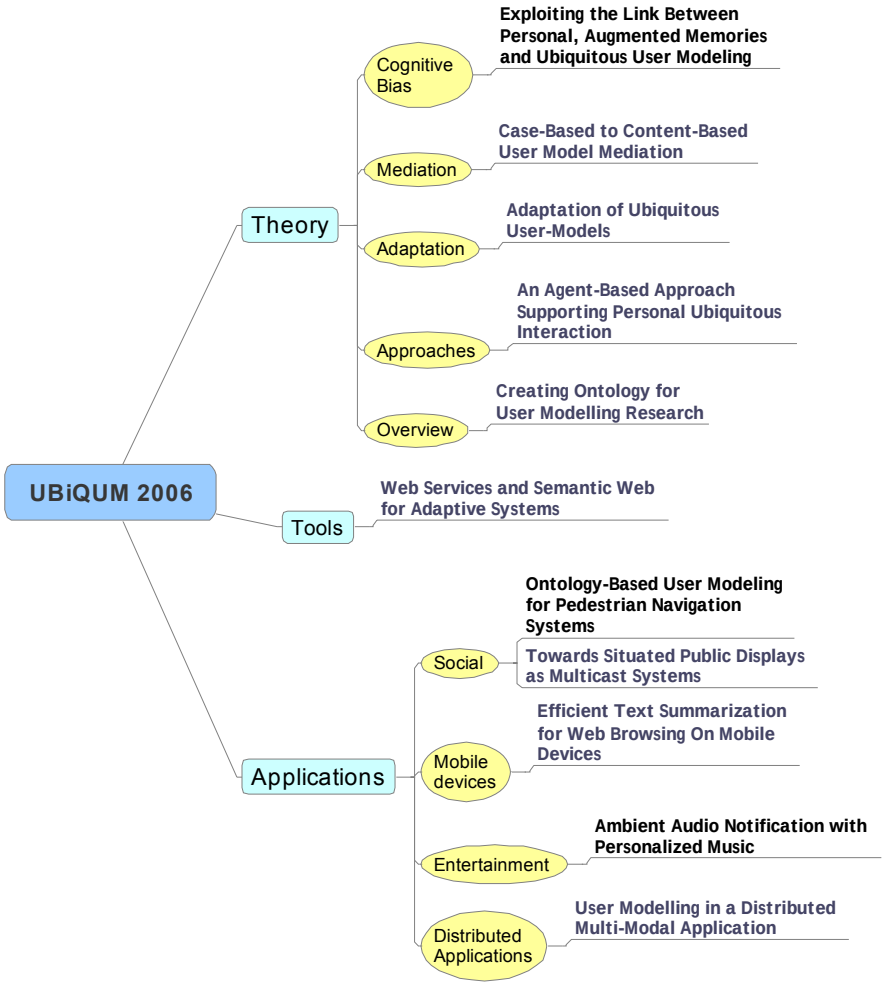


Fig. 1. Mind Map of the main topics of UBIQUM 2006 workshop

The most challenging was meta-level labels definition process and the problem of location as many papers may be attributed to several branches. Ontological engineering procedure is subjective but very rewarding. It shows

- the level of understanding of the problem or expertise,
- the research bias,
- the background, etc.

We suppose that any ontology is better than none. As any roadmap is helpful, but… Wrong roadmap is dangerous. The analysis of Fig. 1 may help to create some conclusions:

- The workshop programme was well balanced between "theory" and "applications".
- The branch "tools" was underrepresented may be because UBIQUM applications are very specific now and are not general enough for creating more or less universal tools.
- Small number of papers doesn't give any chance to create representative groups for more deep analysis.

The authors will be very thankful to any other comments or interpretations of Fig.1. Alternative mind maps of the same workshop are very welcome!

It will be a challenging and exciting work of comparing this subjective view with other ones or analyzing several workshops on the same topic.

4 Analyzing UBIDEUM Call For Papers

Another research interest for us deals with very common work that is done rather often by any researcher – it is looking through call for papers. Normally this process pursuits several goals, e.g. answering the questions like

- Does this CfP match my interests?
- What is the level of the conference?
- Does the event deadline give any chance to write something substantial and relevant to the topic list?
- Is the time and place nice?

The four questions above may be answered after unconscious assessment and evaluation based on level of individual expertise in the domain. The task seems to be totally impossible for the newcomers and the beginners. We suppose that one of the way to facilitate this evaluation procedure may be the mind mapping of CfP. We tried to do it for UBIDEUM'2007. Next paragraph is borrowed from the CFP.

" Topics of interest include, but are not limited to:

- Generic user modelling in mobile and ubiquitous computing
- Context aware ubiquitous user modelling (in mobile and distributed environments
- Construction and acquisition of distributed user models –
- Semantic web approaches for user modeling (i.e. user model ontologies)
- Privacy, security and trust in decentralized user modelling
- Personalized and adaptive applications and interfaces in decentralized and ubiquitous environments
- Case studies, user experience and evaluation of ubiquitous and decentralized UM approaches
- Distributed architectures and interoperability of personalized applications like recommender systems, adaptive hypermedia, e-learning, adaptive navigation guides, personalized shopping guides, etc.
- Service-oriented architectures for decentralized and ubiquitous user modelling and adaptive systems
- Dynamic changes and their implications on the adaptive services in decentralized and ubiquitous environments
- Knowledge modelling, integration and management for personalization in constrained environments
- Reasoning methods in constrained environments
- Personalized content authoring, delivery and access in mobile environments
- Personalized multimedia applications
- Ubiquitous access to personalized applications
- Challenges for user personalization in mobile/distributed environments"

Creating a mind map representing ontology of the future workshop was even more sophisticated activity that designing of ontology on Fig.1. The level of synonym and excessiveness was higher. We have two assumptions on that. First, it is understandable as the purpose of CfP is to invite the wider scope of researchers - so as topics works as attractors they should cover all possible relevant domains that may bear different names.

But it is well known thesis that science begins of classification, and classification begins from the glossary creating. The Fig. 2 clearly shows that the glossary is under development yet. The second reason for heterogeneity is caused by collaborative character of work when collective decision is taken by disjunction not conjunction algorithm.

There are a lot of other considerations born in the process of this mind map design and development. But the mot exciting will be the assessment done by the UBIDEUM'2007 participants. It will be interesting to create the same

mind-map after the conference and to compare the “to be” mind map of CfP and “as is” mind map of accepted contributions.

The other step may be done in comparison of different mind-maps of CfP done by other participants.

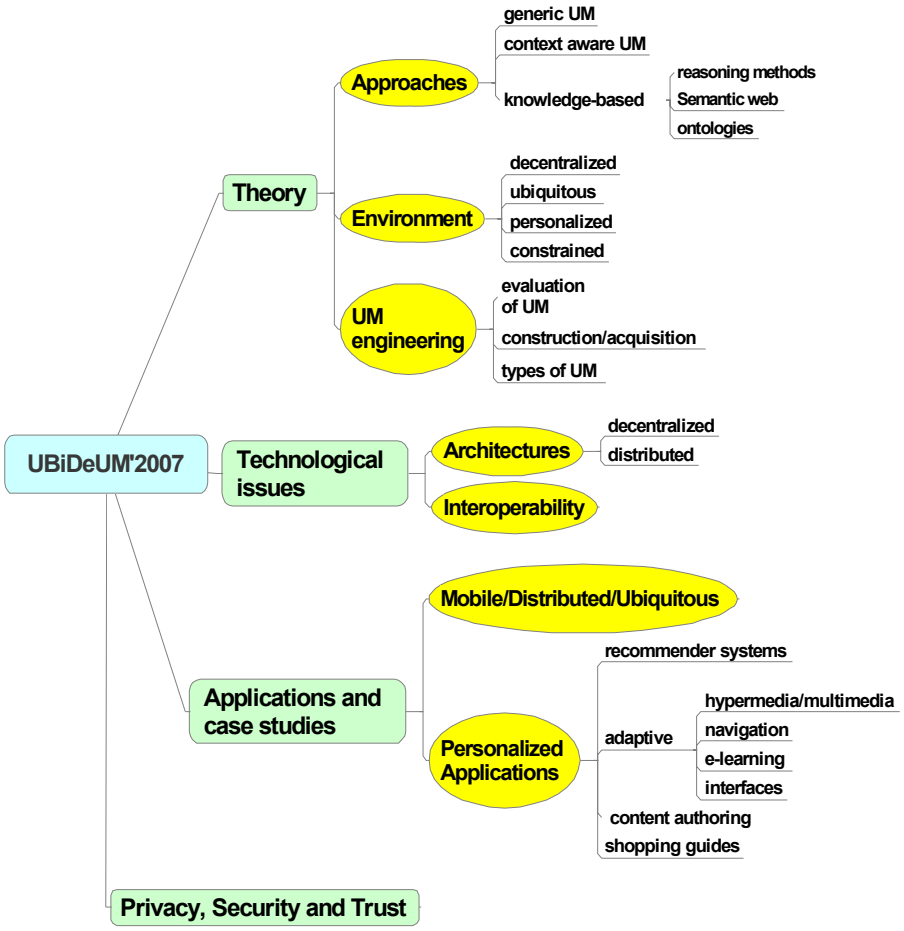


Fig.2 Mind map of UBIDEUM’2007 topics

Discussion

The quintessential issue of such upper-level ontology is not identification of the lower level concepts that correspond to the individual approaches, but the working out and verbalizing the meta-level concepts that would help generalizing the main concepts and paradigms of ubiquitous and decentralized UM methodology. The role of such map in any of research fields is manifold. First, such meta-ontology is an important uniform framework to structure this science field in general. Second, it can be used as a roadmap for the beginners and as a skeleton for teaching. Since the field is large and really ill-structured, upper-level ontology is useful as an indexing tool for the learning material.

And may be its main goal is knowledge sharing and understanding within the research community. The presented ontologies don’t pretend to be ultimate they only put first stones into the foundation of mutual understanding on the research we are doing together.

The similar approach may be applied to KDS conferences, and working out of KDS’ ontology may be done just on the site of the conference.

Acknowledgements

The work reported in this paper is supported by grant 06-01-81005 of RFBR and from Academy of Finland.

Bibliography

1. ACM Computing Classification System http://www.acm.org/class
2. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modelling and User-Adapted Interaction, 6 (2-3), (1996) 87-129
3. Brusilovsky P., Yudelson M., Gavrilova T.:Towards User Modeling Meta Ontology// in Lecture Notes in Artificial Intelligence (LNAI 3538) Ed. Ardissono L., Brna P. & Mitrovic A. Proceedings of 10th International conference User Modeling UM 2005, Springer (2005) 48-452
4. Busan, T.:Mind Map Handbook. Thorsons (2005)
5. Gavrilova, T, .Brusilovsky P., Yudelson M., Puuronen S.: Creating Ontology for User Modelling Research // Workshop "Ubiquitus User modeling" on European Conference on Artificial Intelligence, Riva del Garda, Italy (2006) 11-15
6. Gavrilova, T., Laird, D.: Practical Design Of Business Enterprise Ontologies // In Industrial Applications of Semantic Web (Eds. Bramer M. and Terzyan V., Springer (2005) 61-81
7. Kobsa, A. User Modelling: Recent Work, Prospects and Hazards // In: M. Schneider-Hufschmidt, T. Kühme and U. Malinowski, eds. (1993): Adaptive User Interfaces: Principles and Practice. Amsterdam: North Holland Elsevier.
8. Fischer, G.: User Modelling in Human-Computer Interaction. User Model. User-Adapt. Interact. 11(1-2) (2001) 65-86
9. [Wiki, 2007] http://en.wikipedia.org/wiki/Mind_map
10. [Swartout et al., 1997] Swartout, B., Patil, R., Knight, K., Russ, T. Toward Distributed Use of Large-Scale Ontologies, Ontological Engineering. AAAI-97 Spring Symposium Series, 1997, 138-148.

Author information

Tatiana Gavrilova – *Saint-Petersburg State Polytechnic University Intelligent Computer Technologies Dpt. 195251, Politechnicheskaya, 29, St. Petersburg, Russia + 7 (812) 550-40-73; e-mail: gavr@csa.ru*

Seppo Puuronen – *University of Jyväskylä, P.O. Box 35, FIN-40014, Jyväskylän yliopisto, Finland; e-mail: sepi@cs.jyu.fi*

ОНТОЛОГИЧЕСКИЙ ВЗГЛЯД НА ТЕОРИЮ АВТОМАТОВ

Сергей Кривой, Людмила Матвеева, Елена Лукьянова, Ольга Седлецкая

Аннотация: *Рассматривается краткая онтология теории автоматов, которая строится на принципах тип автомата – язык, распознаваемый этим автоматом, – приложения. Предлагаемая онтология не претендует на полноту, поскольку теория автоматов стала столь обширной, что обозреть все ее аспекты является трудной задачей.*

УДК 51.681.3

1. Введение

Одной из важных проблем в современном техническом и программном обеспечении компьютеров является проблема корректного проектирования и верификации полученного продукта. Если проблема проектирования относится к трудно формализуемым проблемам (а, следовательно, к трудно автоматизируемым), то проблема верификации является трудной в связи с тем, что она в себе аккумулирует множество разных задач из смежных областей, чем и объясняется ее сложность. Одной из таких смежных областей является теория автоматов, которая нашла и находит все новые и новые приложения в частичном решении проблем проектирования и верификации. В частности, если речь

заходит о верификации свойств реактивных систем, то теория автоматов играет здесь ключевую роль, поскольку такие важные проблемы, как проблема распознавания свойств, достижимости определенного состояния или множества состояний и пустоты акцептируемого языка разрешимы в теории автоматов для большинства типов автоматов.

В данной работе построена краткая онтология теории автоматов исходя из таких зависимостей ***автомат – язык, распознаваемый этим автоматом, – приложения.*** В связи с ограниченным объемом данной работы, кратко рассматриваются только основные свойства автоматов и их приложения.

2. Конечные автоматы над конечными словами

Теория конечных автоматов над словами конечной длины является законченной теорией в том смысле, что все основные теоретические задачи этой теории решены. Основные результаты, которые получают в этой области, - это результаты прикладного характера, т. е. результаты применения методов теории автоматов к решению задач в конкретных прикладных областях. Однако, не смотря на это, теория конечных автоматов оказала влияние на дальнейшее развитие общей теории автоматов. Это проявляется в том, что появились многочисленные вариации понятия конечного автомата такие, как конечные автоматы над бесконечными словами [3, 12], временные автоматы [2], гибридные автоматы [9], автоматы над деревами [10] и т. д. и т. п.

Основным понятием, от которого строится онтология, является понятие конечного автомата над словами конечной длины в конечном алфавите. Среди таких автоматов наиболее употребительными являются три типа автоматов: *автоматы Мили, автоматы Мура и X –автоматы (или автоматы без выходов)*. Перейдем к определению этих типов автоматов.

Автоматы Мили. Пусть $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_m\}$ – конечные алфавиты, т. е. конечные множества попарно различных элементов, которые называются символами или сигналами.

Определение 1. Пятерка *(A, X, Y, f, g)* называется автоматом Мили, если она состоит из множества *A* состояний автомата, алфавита входных символов *X*, алфавита выходных символов *Y,* функции переходов $f : A \times X \to A$ и функции· выходов $g: A \times X \to Y$. Алфавиты *X* и *Y* называют соответственно входным и выходным алфавитами символов или сигналов автомата.

Как правило, автомат обозначают символом множества его состояний, т. е. *A = (A, X, Y, f, g).* Если *f(a,x) = a'*, то говорят, что автомат *A* под действием входного сигнала $x \in X$ переходит в состояние *a'*, или что сигнал *x* переводит автомат *A* из состояния *a* в состояние *a'*. Если *g(a,x) = y*, то говорят, что автомат *A* преобразует в состоянии *a* входной сигнал $x \in X$ в выходной сигнал $y \in Y$. Функционирование автомата описывается таким образом. На вход автомата подается сигнал (символ) $x \in X$, под действием которого в соответствии с текущим состоянием автомата и функциями переходов и выходов изменяется состояние автомата и на его выходе появляется выходной сигнал (символ) $y \in Y$. Если на вход автомата подавать слово (последовательность входных символов), то на выходе также появляется слово (последовательность выходных символов). В этом случае автомат *A* можно рассматривать как алфавитный преобразователь информации, который отображает слова полугруппы *F(X)* в слова полугруппы *F(Y)*. Последовательность входных сигналов можно рассматривать как функцию натурального аргумента - дискретного автоматного времени. Это обстоятельство позволяет рассматривать автомат как дискретную динамическую систему, которая изменяет свои состояния во времени под действием внешних и внутренних факторов.

Автомат *A* называется инициальным, если в множестве его состояний выделено некоторое состояние a_0, которое называется начальным состоянием. При функционировании инициального автомата считается, что в начальный момент времени (перед подачей на его вход некоторого слова $p \in F(X)$) автомат находится в начальном состоянии a_0.

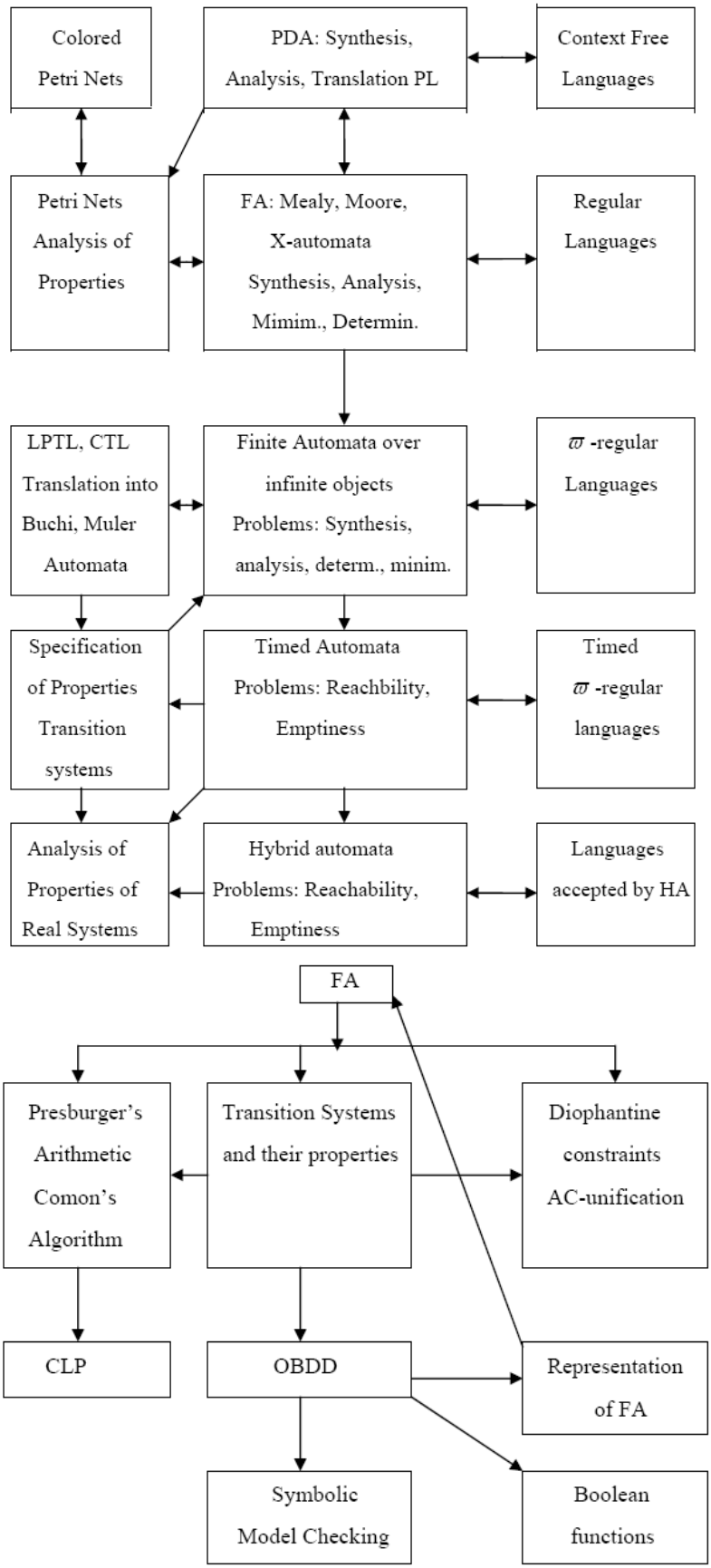

Colored Petri Nets
PDA: Synthesis, Analysis, Translation PL
Context Free Languages
Petri Nets Analysis of Properties
FA: Mealy, Moore, X-automata Synthesis, Analysis, Mimim., Determin.
Regular Languages
LPTL, CTL Translation into Buchi, Muler Automata
Finite Automata over infinite objects Problems: Synthesis, analysis, determ., minim.
ϖ -regular Languages
Specification of Properties Transition systems
Timed Automata Problems: Reachbility, Emptiness
Timed ϖ -regular languages
Analysis of Properties of Real Systems
Hybrid automata Problems: Reachability, Emptiness
Languages accepted by HA
FA
Presburger's Arithmetic Comon's Algorithm
Transition Systems and their properties
Diophantine constraints AC-unification
CLP
OBDD
Representation of FA
Symbolic Model Checking
Boolean functions

Рис. 1

Автомат $A = (A, X, Y, f, g)$ называется конечным, если все три множества A, X и Y конечны и бесконечным, если хотя бы одно из этих множеств бесконечно.

Автомат называется полным или полностью определенным, если функция переходов и выходов полностью определены и частичным, если хотя бы одна из этих функций частичная.

Автоматы у которых компонента f не является функцией, а является некоторым отношением, т. е. в таких автоматах не выполняется условие однозначности перехода, называются недетерминированными. Если же отношение f есть функцией, то автомат называется детерминированным. Следовательно, для детерминированного автомата A с начальным состоянием a_0 однозначно находится состояние b, в которое автомат переходит под действием слова $p \in F(X)$. А в недетерминированном автомате таких состояний может быть не одно а несколько. Ясно, что класс детерминированных автоматов Мили является подклассом класса недетерминированных автоматов.

Автоматы Мура. Автоматы Мура есть частным случаем автоматов Мили.

Определение 2. Автомат (A, X, Y, f, g) называется автоматом Мура, если его функция выходов $g(a,x)$ выражается через функцию переходов $f(a,x)$ при помощи уравнения $g(a,x) = h(f(a,x))$, где $h: A \to Y$. Функция h называется функцией отметок автомата, а ее значение $h(a)$ на состоянии a называется отметкой этого состояния.

Несмотря на то, что автоматы Мура есть частным случаем автоматов Мили, в теории автоматов они являются объектом отдельного изучения, поскольку в ряде случаев их специфические свойства дают возможность строить более содержательную теорию, чем теория автоматов Мили.

X-автоматы. X-автоматом называется четверка (A, X, f, F), а если X-автомат инициальный, то пятерка (A, X, f, a_0, F), где $f: A \times X \to A$ - функция переходов, $F \subseteq A$ - некоторое подмножество состояний автомата, элементы которого называются заключительными состояниями, а $a_0 \in A$ – начальное состояние автомата.

3. Краткая онтология конечных автоматов

На рис.1 представлена краткая онтология теории автоматов, которая приведена в виде графа. Дуги этого графа связывают различные области, которые примыкают к той или иной теории автоматов или к ее приложениям.

4. Конечные автоматы над бесконечными словами

Пусть X – конечный алфавит. ϖ -языком называется произвольное подмножество X^{ϖ} - всех бесконечных слов в алфавите X. ϖ -автомат является конечным объектом для акцептации· ϖ -языков над некоторым алфавитом X. Существует несколько разновидностей ϖ -автоматов. Основными среди них есть автоматы Бюхи и Мюлера.

4.1. Автоматы Бюхи и автоматы Мюлера

Определение 3. Автоматом Бюхи A называется объект (A, X, f, A_0, F), где X – конечный входной алфавит, A - множество состояний автомата, $A_0 \subseteq A$ - множество начальных состояний, $f \subseteq A \times X \times A$ - множество дуг и $F \subseteq A$ - множество заключительных состояний. Объект (A, X, f, A_0) называется таблицею переходов.

Автомат стартует в одном из начальных состояний и если $(s, x, s') \in f$, то автомат может изменить сове состояние из s на s' путем чтения входного символа $x \in X$. Для слова $\sigma = x_1 x_2 x_3 \ldots$ в алфавите X

будем говорить, что $r = s_0 x_1 s_1 x_2 s_2 x_3 \ldots$ является трассой автомата A над σ, которая соединяет $s_0 \in A_0$ и $(s_{i-1}, x_i, s_i) \in f$ для всех $i \geq 1$. Для трасы r, множество $inf(r)$ состоит из состояний $s \in A$ таких, что $s = s_i$ для бесконечного числа раз, $i \geq 0$. Трасса r автомата A над словом $\sigma \in X^{\varpi}$ называется акцептирующей трассой, тогда и только тогда, когда $inf(r) \cap F \neq \varnothing$. Другими словами, трасса r является акцептирующей тогда и только тогда, когда некоторое состояние из множества F повторяется бесконечно часто на трассе r. Язык $L(A)$ называется акцептированным автоматом A, если он состоит из слов σ ϖ таких, что автомат A имеет трассу r, которая акцептирует σ.

ϖ -язык называется регулярным ϖ -языком, если он акцептируется некоторым автоматом Бюхи.

Пусть $L(A) = \{p \in X^{\varpi} \mid A$ акцептирует $p\}$ - ϖ -язык, который акцептируется некоторым автоматом A. Если $L = L(A)$ для некоторого автомата Бюхи A, то говорят, что язык L является Бюхи акцептируемым.

Обобщением автоматов Бюхи относительно множества заключительных состояний являются автоматы Мюлера.

Определение 4. Автоматом Мюлера над алфавитом X называется объект $A = (A, X, f, a_0, F)$, где X - конечный алфавит, A - конечное множество состояний, $a_0 \in A$ – начальное состояние, $f \subseteq A \times X \times A$ - отношение переходов, $F \subseteq B(A)$ - множество подмножеств множества заключительных состояний. Говорят, что трасса $\sigma = s_0 x_1 s_1 x_2 s_2 x_3 \ldots$ автомата является эффективной, если некоторые состояния этой трассы появляются бесконечное число раз и являются состояниями из множества F. Автомат А акцептирует слово $p \in X^{\varpi}$, если трасса, которая соответствует слову p, является эффективной. ϖ -язык $L \subseteq X^{\varpi}$ называется акцептованным автоматом Мюлера, если он состоит из всех ϖ -слов, которые акцептирует этот автомат.

4.2. Свойства регулярных ϖ -языков

Все результаты для ϖ -языков и ϖ -автоматов приведены в таблице 1. Приведенные в этой таблице результаты означают замкнутость соответствующих операций по отношению в данному классу автоматов

Таблица 1

Класс ϖ языков	Операции
МА = ВА = ДМА $\cup$ ДВА	Объединение, пересечение, дополнение Объединение, пересечение

5. Временные автоматы

Пусть B - множество переменных часов, значения которых есть неотрицательными действительными числами $D+$. Множество временных ограничений $C(B)$ определяется таким образом:

a) все элементы, принадлежащие к $C(B)$, являются неравенствами и имеют вид $y \prec c$ и $c \prec y$, где $\prec$ означает < или $\leq$, а c – неотрицательное рациональное число;

b) если ϕ и ψ принадлежат к $C(B)$, то $\phi \wedge \psi \in C(B)$.

Заметим, что если B включает k часов, то каждое временное ограничение определяет некоторое выпуклое множество k-мерного евклидового пространства. Следовательно, если две точки удовлетворяют временному ограничению, то и все точки, которые лежат на отрезке, который соединяет эти точки, также удовлетворяют этому временному ограничению.

Определение 5. Временным автоматом (ВА) *A* называется шестерка *(X, A, A_0, B, I, T)*, где *X* – конечный алфавит ВА, *A* – конечное множество состояний, $A_0 \subseteq A$ – множество начальных состояний, *B* – множество часов, $I : A \rightarrow C(B)$ – отображение множества состояний во временные ограничения, которое называется инвариантами состояний, $T \subseteq S \times X \times C(B) \times 2^B \times A$ – множество переходов.

Каждая пятерка (a, x, ϕ, λ, a') соответствует переходу из состояния a в состояние a', который отмечен символом x $\in$ X. Ограничение ϕ определяет момент времени, когда этот переход станет возможным, а значение часов из множества $\lambda \in$ B сбрасываются в ноль при выполнении этого перехода**.**

5.1. Модель временного автомата

Моделью ВА *A* служит граф переходов *T(B) = (X, V, V_0, R)* с бесконечным числом вершин. Каждая вершина из *V* представляет собой пару *(a, v)*, которая состоит из состояния $a \in A$ и значения часов $v : B \rightarrow D+$, которые они принимают в множестве неотрицательных рациональных чисел. Множество начальных вершин есть таким: $V_0 = \{(a,v) : a \in A_0 \wedge \forall y \in B[v(y) = 0]\}$.

Для определения переходов в *T(B)* необходимо ввести некоторую нотацию. Для $\lambda \subseteq B$ определим $v[\lambda = 0]$ как значения часов, в которых значения совпадают с *v* для часов из $B \setminus \lambda$ и приписывает значение 0 всем часам из λ.

Для $d \in D+$ определим $v + d$ как значение часов, которые присваиваются всем часам из $y \in B$ значение $v(y) + d$, а значение часов $v - d$ определяется аналогично.

Из определений следует, что ВА имеет два основных типа переходов:

- переход по задержке *d* соответствует определенному времени в том случае, когда автомат находится в некотором состоянии *a*. При этом пишут *(a, v){d} (a, v + d)*, где $d \in D+$, в том случае, если для всех $0 \leq e \leq d$ инвариант *I(a)* истинен для $v + e$;
- переход по действию соответствует выполнению перехода из множества *T*. В этом случае пишут, что *(a, v){x}(a', v')*, где $x \in X$, в том случае, если существует такой переход *(a, x, ϕ, λ, a')*, что v удовлетворяет ϕ и $v' = v[\lambda = 0]$.

Отношение переходов *R* для *T(B)* получается путем объединения множества переходов по задержке с множеством переходов по действию. Запись *(a,v)R(a', v')* или *(a, v){x}(a', v')* означает, что существуют такие элементы *a''* и *v''*, что *(a, v){d} (a'', v''){x} (a', v')* , для некоторого $d \in D+$ и $x \in X$.

5.2. Временные языки

Пусть $X = \{x_1,\ldots,x_n\}$ алфавит и *R* – множество рациональных чисел. Временной последовательностью $\tau = \tau_1 \tau_2 \ldots$ называется бесконечная последовательность значений времени $\tau \in R$ такая, что $\tau_i > 0$ и удовлетворяет таким условиям:

(1) Монотонность: τ растет строго монотонно, т. е. $\tau_i < \tau_{i+1}$ для всех $i \geq 1$;

(2) Прогресс: для каждого $t \in R$ существует некоторое $i \geq 1$ такое, что $\tau_i > t$.

Временным словом в алфавите *X* называется пара (σ, τ), где $\sigma = \sigma_1 \sigma_2 \ldots$ является бесконечным словом над *X*, а τ есть временной последовательностью. Временным языком над алфавитом *X* называется множество временных слов над *X*.

Временное слово (σ, τ) называется входным словом автомата и в каждый момент времени *i* входной символ σ_i определяет момент времени τ_i.

Операции на временных языках такие как объединение, пересечение и дополнение определяются так же, как и для регулярных языков.

Для временного языка *L* в алфавите *X* оператор *Untime(L)* представляет собой проекцию временного слова на его первую компоненту, т. е . *Untime(L)* есть ϖ -языком, который включает все слова σ такие, что $(\sigma, \tau) \in L$ для некоторой временной последовательности τ . Основные результаты для языков и операций над ними приведены в таблице 2.

Таблица 2

Класс временных языков	Операции
ТМА = ТВА	Объединение, пересечение
$\cup$	
ДТМА	Объединение, пересечение, дополнение
$\cup$	
ДТВА	Объединение, пересечение

6. Гибридные автоматы

Понятие гибридного автомата является обобщением понятия временного автомата. Гибридный автомат (ГА) представляет собой комбинацию дискретной динамики конечного автомата и непрерывной динамики динамической системы. ГА является удобной математической моделью многих реальных систем таких, как цифровые компьютерные системы, которые взаимодействуют с аналоговым окружением в реальном времени. В частности, при помощи ГА моделируются распределенные процессы со смешанным временем, протоколы управления и контроля в промышленных изделиях, движущиеся объекты, роботы и т. п.

Основными проблемами в теории ГА есть две проблемы:

- проблема достижимости и более общая
- **проблема пустоты** ϖ -языка.

Приведем основные определения и результаты в теории ГА. Пусть $D_{\geq 0}$ означает множество неотрицательных действительных чисел, т. е. $D_{\geq 0} = \{x \in D \mid x \geq 0\}$.

Прямоугольные области. Для натурального числа $n > 0$, $n \in N$ подмножество из D^n называется областью или регионом. Ограниченная и замкнутая область называется компактной. Область $R \subseteq D^n$ называется прямоугольником, если она представляет собой декартовое произведение интервалов (возможно и неограниченных), предельные точки которых являются рациональными числами. R_i означает проекцию *R* на *i*-тую координату, т. е. $R = R_1 \times R_2 \times \ldots \times R_n$. Множество всех прямоугольных областей в D^n обозначается R^n **.**

Определение 6. *n*-мерным прямоугольным автоматом (ПА) *A* называется такая девятка *A = (G, X, init, inv, flow, pre, post, jump, obs)*, где *G = (V, E)* – конечный ориентированный граф, *X* – конечный алфавит символов, для наблюдения, трех функций $init : V \to R^n$, $inv : V \to R^n$, $flow : V \to R^n$, которые отмечают вершины графа *G,* четырех функций pre: $E \to R^n$, $post : V \to R^n$, $jump : E \to 2^{\{1,2,\ldots,n\}}$, $obs: E \to X$, которые помечают дуги графа *G*.

ПА может иметь так называемые ε -переходи (пустые переходы). *n*-мерный прямоугольный автомат с ε переходами отличаются от приведенного выше тем, что функция *obs* есть такой: $obs : E \to X^{\varepsilon}$, где $X^{\varepsilon} = X \cup \{\varepsilon\}$.

Функция *init* определяет множество начальных состояний ПА. Если дискретное состояние начинается в вершине $v \in V$, то непрерывное состояние должно начинаться в начальной области *init(v)*.

Функции *pre, post, jump* ограничивают поведение ПА во время выполнения переходов по дугах графа *G*. Дуга *e = (v, w)* может быть пройдена только когда дискретное состояние остается в вершине *v* и непрерывное состояние переходит в область *pre (v)*. Для каждого числа *i* множество вершин, в которые возможен переход (множество *jump set)*, *i*-ой координате непрерывного состояния недетерминированным образом присваивается новое значение из интервала *post (e).* Для каждого $i \notin$ *jump (e)*, *i*-тая координата непрерывного состояния не изменяется и остается в *post (e)*.

Функция наблюдения *obs* идентифицирует каждое ребро перехода помеченного символом наблюдения из *X* или из X^{ε}.

Функции *inv* и *flow* ограничивают поведение ПА во время переходов. До тех пор, пока дискретное состояние остается в вершине *v*, непрерывное состояние недетерминировано следует плавной траектории (C^{∞}) внутри инвариантной области *inv(v),* чье начальное время остается в текущей области *flow (v)*.

ПА с ε-переходами может двигаться по ε-дугах во время выполнения переходов.

Если в определении ПА заменить области произвольными линейными областями, то полученный автомат называется линейным гибридным автоматом. Прямоугольные автоматы составляют подкласс класса вех линейных гибридных автоматов, в которых все определяемые области прямоугольные.

Инициализация и ограниченный недетерминизм. ПА *A* называется инициальным, если для каждой дуги *e=(v, w)* из *A* и каждой координаты $i \in [1,2,\ldots,n]$ и $flow(v)_i \neq flow(w)_i$ имеет место включение $i \in jump(e)$**.**

Отсюда следует, что каким бы образом не изменялась *i*-тая непрерывная координата инициального ПА, которая задана *flow* функцией, то ее значение недетерминированным образом реинициализируется в соответствии с *post* функцией.

ПА *A* называется ограничено детерминированным, если

(1) для каждой вершины $v \in V$ области *init(v)* и *flow(v)* ограничены и

(2) для каждой дуги e ∈ E и каждой координаты i ∈ [1,…,n] з того, що i ∈ jump(e) вытекает, что интервал post(e) ограничен.

Заметим, что из условия ограниченности недетерминизма не следует условие конечности ветвлений в состоянии. Отсюда только следует, что время переходов по дугах в ограниченной области является ограниченным.

7. Приложения автоматов для верификации систем

Конечные автоматы с успехом используются для моделирования параллельных и взаимодействующих реактивных систем. При таком моделировании или множество состояний автомата, или символы его входного алфавита представляют состояния моделируемой системы. Основным преимуществом при таком использовании автоматов для верификации систем есть то, что как модель самой системы, так и ее спецификация представляются одинаково. В такому случае модель Крипке непосредственно соотносится с ϖ-автоматом (Б'юхи или М'юлера), все состояния которой являются заключительными, а множество возможных поведений системы задается ϖ-языком *L(A)*, который акцептируется соответствующим автоматом *A*. При этом существует алгоритм, который транслирует произвольную формулу темпоральной пропозициональной логики в ϖ-автомат Если *AP* – множество атомарных высказываний, то в частности, модель Крипке *(S, R,* S_0*, f)*, где $f : S \to 2^{AP}$, можно преобразовать в автомат $A = (A \cup a_0, X, Q, a_0, A \cup a_0)$, у которого входной алфавит есть булеаном множества атомарных высказываний $X = 2^{AP}$. При

этом для каждой пары состояний *a, a′* $\in$ *A* отношение *Q* включает тройку *(a,x,a')* тогда и только тогда, когда *(a,a')* $\in$ *R* и *x = f(a')*, причем $(a_0, x, a) \in Q$ в том и только в том случае, когда $a_0 \in S_0$ и *x = f(a)*, где *R* – отношение следования в модели Крипке.

8. Примеры некоторых свойств и их спецификаций

Рассмотрим некоторую гипотетическую реактивную систему. Свойства таких систем делят на два класса:

- **свойства безопасности (*safety properties*),** которые говорят о том, что нечто нежеланное не появится никогда**;**
- **свойства живучести (*liveness properties*),** которые говорят о том, что нечто хорошее в конце концов появится в системе.

Свойства безопасности выражаются, как правило, с помощью формул темпоральной логики вида $\Box \neg p$, где формула *p* характеризует не желаемое событие (состояние) в системе. Примером свойства безопасности может служить свойство взаимного исключения (*mutual exclussion*): $\Box (\neg p \vee \neg q)$, где формулы *p* и *q* характеризуют состояния, в которых система никогда не может находится одновременно. Примером свойства живучести может служить свойство справедливости (*fairness*), которое говорит о том, что когда к системе поступает запрос, то на него когда-нибудь будет получен ответ.

Расширим этот набор свойств еще некоторыми свойствами, которые принадлежат к классу свойств живучести. Эти свойства выглядят так:

- гарантия (guarantee) говорит о том, что некоторое событие появится хотя бы один раз, однако не гарантирует его повторения. Это свойство выражается ЛПТЛ-формулой $\Diamond p$;

- объязательство (obligation) говорит о том, что формула *p* должна выполнятся всегда или формула *q* выполняется в том же состоянии, что и формула *p*. Это свойство выражается ЛПТЛ-формулой $\Box p \vee \Diamond q$;

- реакция (response) говорит о том, что событие, которое описывается формулой *p*, появляется бесконечное число раз. Это свойство выражается ЛПТЛ-формулой $\Box \Diamond p$;

- настойчивость (persistense) говорит о том, что после некоторой задержки наступает стабилизация события·, которое описывается формулой *p*. Это свойство выражается ЛПТЛ-формулой $\Diamond \Box p$;

- реактивность (reactivity) представляет собой дизъюнкцию свойств реакции и настойчивости. Это свойство выражается ЛПТЛ-формулой $\Diamond \Box p \vee \Box \Diamond p$;

- безусловная справедливость (unconditional fairness) говорит о том, что событие, которое описывается формулой *q*, появляется бесконечное число раз независимо от условия *p*. Это свойство выражается ЛПТЛ-формулой $\Box \Diamond p$;

- слабая справедливость (weak fairness) говорит о том, что когда формула *p* все время истинная, то формула *q* должна быть истинной бесконечно часто. Это свойство выражается ЛПТЛ-формулой $\Box p \rightarrow \Box \Diamond q$;

- сильная справедливость (strong fairness) говорит о том, что когда формула *p* истинная бесконечно часто, то формула *q* должна быть истинной тоже бесконечно часто. Это свойство выражается ЛПТЛ-формулой $\Box \Diamond p \rightarrow \Box \Diamond q$.

Литература

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. -М. :Мир.-1979. - 535 с.
2. Alur R., Dill D.L. A theory of timed automata. - Theoretical Computer Science. -1994. -126. - PP. 183-235.
3. Thomas W. Automata on infinite objects. Handbook on theoretical computer science. - 1990. - PP. 135-191.
4. Годлевский А.Б., Кривой С. Л. Трансформационный синтез эффективных алгоритмов с учетом дополнительных спецификаций. Кибернетика, - 1986. - N 6. - С.34 - 43.
5. Глушков В.М. Абстрактная теория автоматов. - Успехи математических наук. -1961. - 16. - вып. 5. - С. 3-62.

6. Глушков В.М. Синтез цифровых автоматов. - М: Физматгиз. - 1962. - 476 с.
7. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра, языки, программирование. -Киев: Наукова думка. -1985. 327с.
8. Perrin D. Finite automata. In Handbook of Theoretical Computer Science. vol. 2, -Elsevier. -1990. -PP. 1-58.
9. Henzinger T.A., Kopke P. W, Puri A., Varaiya P. What's Decidable About Hybrid Automata? In the Proceed. of the 27-th Annual ACM Symposium on Theory of Computing (STOC 1995). - 1995. - PP. 373-382.
10. Comon H. Constraint solving on terms: Automata techniques (Preliminary lecture notes). - Intern. Summer School on Constraints in Computational Logics: Gif-sur-Yvette, France, September 5-8. -1999. - 22 p.
11. Капитонова Ю.В., Кривой С. Л., Летичевский А. А., Луцкий Г.М. Лекции по дискретной математике. БХВ: Санкт-Петербург, 2004, 624 с.
12. Трахтенброт Б. А., Барздинь Я. М. Конечные автоматы (Поведение и синтез). -М.: Наука. -1970. - 400 с.
13. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. -М.: Мир. -1973. - 256 с.
14. Arnold A. Finite Transition Systems: Semantics of Communicating Systems. -Paris: Prentiuce Hall. -1994. - 177 p.
15. Ben-Ari M. Mathematical Logic for Computer Science. Springer Verlag London Limited. - 2001.-305 p.
16. Emerson E.A. Temporal and modal logics. Handbook of Theoretical Computer Science: Elsevier. - vol. B. -1990. - PP.995-1072.
17. Peterson G.L. Myths about the mutual exclusion problem. - Information Processing Letters, - 1981. - v.12.-N 3. - P.115-116
18. Wolper P. Temporal logic can be more expressive. - Information and Control. -v. 99. -1983. -P. 56 - 72.
19. Clarke E.M., Grumberg Jr. O., Peled D. Model Checking. - The MIT Press: Cambridge, Massachusetts, London, England. -2001. -356 p.

Информация об авторах

С. Л. Кривой, Л. Е. Матвеева *– Институт кибернетики им. В. М. Глушкова НАН Украины, Киев, Украина, e-mail: krivoi@i.com.ua*

Елена А. Лукьянова *– Таврический национальный университет им. В.И. Вернадского, Симферополь, Украина,*

Ольга Седлецкая *– Ченстоховский политехнический институт, Ченстохов, Польша, e-mail: olga@icis.pcz.pl*

ONTOLOGY-DRIVEN INTRUSION DETECTION SYSTEMS

Vladimir Jotsov

Abstract*: We consider and analyze different types of ontologies and knowledge or mataknowledge connected to them aiming at realization in contemporary information security systems (ISS) and especially the case of intrusion detection systems (IDS). Human-centered methods INCONSISTENCY, FUNNEL, CALEIDOSCOPE and CROSSWORD are algorithmic or data-driven methods based on ontologies and interacting on a competitive principle. They are controlled by a synthetic metamethod SMM. It is shown that the data analysis in the field frequently needs an act of creation especially if it is applied in a knowledge-poor environment. It is shown that human-centered methodsare very suitable for resolution of the quoted tasks.*

Introduction

Contemporary information security systems (ISS) and especially those Internet-based are primarily based on the usage of intelligent methods. The case of intrusion detection systems (IDS) is machine learning oriented, and some of them are using data mining [1,2]. Such sophisticated technologies are time- and labor-consuming, it is very hard making them satisfy the demands for results convergence and low computational complexity. However

designers and customers accept such difficulties trying to gain from higher security of such decision support applications. Here the base concept is to make a powerful human-centered system combined with firewalls or other passive security tools which complex defends against different groups of intruders. It is obvious that we should introduce different ontologies to support the IDS work or the system will be not enough reliable. In the next two Sections we'll show ontologies usage in different decision support methods and applications in data mining, web mining and/or data warehousing.

Usually ontologies are issued to support methods/applications to probabilistic, fuzzy inference, or uncertainty processing [3-6]. Our research shows other [7], sometimes nonstandard ways that are not defeating the other contemporary research but are making something in addition to well known methods, and so are useful to be combined. The next Section is dedicated to a new self-learning method that constantly searches for knowledge conflicts or its ultimate case-contradictions-and tries to resolve it [8]. This is found to be the best way to self-improvement via the correction of incompleteness or inconsistency. On contrary to other machine learning methods, our proposal is ontology-driven and isn't heuristic by nature. In this case the keyword self-learning is introduced to emphasize the above quoted differences.

In Section 3 different human-centered methods are used to check the truth value of one or group of statements. They are named below in the text: definition of the problem, target question or a *goal* for short, e.g. goal to detect a possible intrusion. We are not trying to elaborate completely automatic systems. Since the first knowledge discovery systems it is seen that making more or less automatic inference system makes it full of heuristics, which restricts its future development. Instead we offer making the machine the best human's advisor which founds some interesting patterns and represents it in an user-friendly manner. Some of similar ideas are used in cognitive graphics but our methods are absolutely different and we prefer to name the filed: human-machine creation.

Application results are considered in Section 4. The above quoted research have been never realized 'all in one' system because of its high complexity. However we used a big variety of method combinations under the synthetic metamethod control. Those allow us make rather effective inference machines.

2. Ontology-Based Machine Learning

Let the strong (classical) negation is denoted by '¬' and the weak (conditional, paraconsistent [9]) negation by '~'. In case of an evident conflict (inconsistency) between the knowledge and its ultimate form–the contradiction–the conflict situation is determined by the direct comparison of the two statements (the *conflicting sides*) that differ one form another just by a definite number of symbols '¬' or '~'. For example A and ¬A; B and not B (using ¬ equivalent to 'not'), etc. η is a type of negation, strong negation in case, and square brackets embrace different words used to represent explicit strong negations in text.

$$\{\eta\}\ [\text{no, not, не, нет}]. \qquad (1)$$

The case of implicit (or hidden) negation between two statements A and B can be recognized only by an analysis of a present ontologies of type (2).

$$\{U\}\ [\eta\text{: A, B}]. \qquad (2)$$

where U is a statement with a validity including the validities of the concepts A and B and it is possible that more than two conflicting sides may be present. Below it is accepted that the contents in the figure brackets U is called *an unifying feature*. In this way it is possible to formalize not only the features that separate the conflicting sides but also the unifying (or common) concepts. For example the intelligent detection may be either automated or of a man-machine type but the conflict cannot be recognized without the investigation of the following ontology (3).

$$\{\text{detection procedures}\}\ [\neg\text{: automatic, interactive}]. \qquad (3)$$

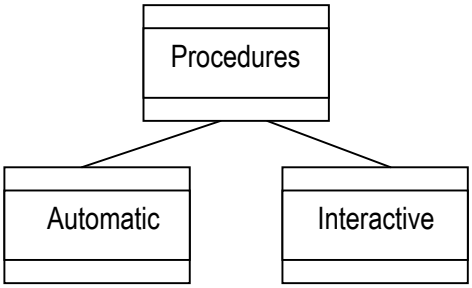


Figure 1. Ontology for a syntactic contradiction

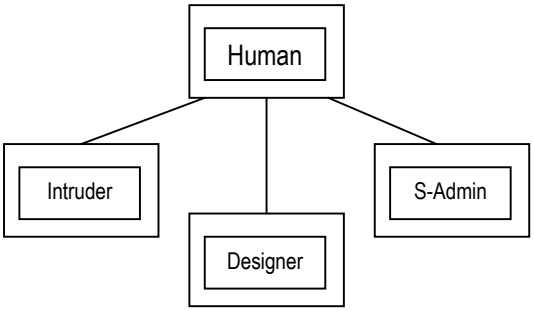


Figure 2. Ontology for a semantic contradiction

Ontologies (1) or (2) describe situations where conflict the sides mutually negate one another. In the majority of situations the sides participate in the conflict only under definite conditions: $\chi_1, \chi_2, \ldots\chi_z$.

$$\{U\}\ [\eta: A_1,A_2,\ldots A_p]\ \ <\tilde{\chi}_1 * \tilde{\chi}_2 * \ldots * \tilde{\chi}_z>. \tag{4}$$

where $\tilde{\chi}$ is a literal of χ, i.e. $\tilde{\chi} \equiv \chi$ or $\tilde{\chi} \equiv \eta\chi$, * is the logical operation of conjunction, disjunction or implication.

The syntactic contradiction ontology is depicted in fig. 1, and the semantic variant is considered in fig. 2. It is obvious that the contradictions are very different but their base ontologies seem quite similar. The reason is the essential part of both conflicts or contradictions from (2) and (3) isn't the ontology knowledge itself but the *metaknowledge* controlling the usage of ontologies or parts of them. The bottom level objects from fig. 1 unconditionally refute each other. We may find some cases where the same system have been automatic one, and after some time it became an interactive system, but this case is so labor consuming that actually we speak about a new, different system.

What is depicted in fig. 2 shows a different situation, concerned with 'IDS-humans' or three major groups of people dealing with IDS: intruders; security experts or designers (designer); security administrators (S-admin). Weak negation is used in case, in the bottom level objects, because the security administrator may be former expert or he may be a designer of another system, and also former hackers may be engaged as experts. The semantic contradiction will appear iff all the conditions are satisfied: T (the same time) and I (the same system) and U (the same person) and P (the same place). Next figures 3 and 4 give more details for the case.

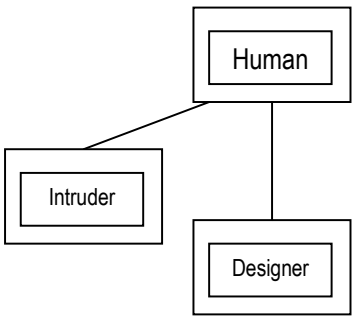


Figure 3. Ontology for conflict situations

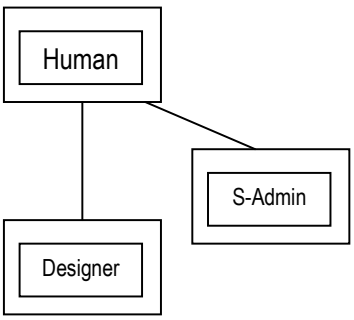


Figure 4. Ontology for contradiction situations

Fig. 3 concerns the part of ontology from fig. 2 when the security administrator is eliminated. Let all the quoted above conditions are satisfied: T (the same time) and I (the same system) and U (the same person) and P (the same place). Still we couldn't define the situation in fig. 3 as a contradiction because the designer may test the IDS system. To resolve this situation we may use knowledge type exclusion and defeasible inference or other well known inference schemes. This is an example of knowledge conflict, not a contradiction, and only additional investigations may result in semantic contradiction.

Fig. 4 shows the semantic contradiction, and if the conditions T∧I∧U∧P are satisfied, then the contradiction appears: 'nobody can occupy both positions'. Thus fig. 2 contains different types of ontology knowledge inside it. The above given examples aim to show that processes based on knowledge conflicts or contradictions couldn't be thoroughly described by ontology knowledge, and using only static situations. We need the dynamic picture to decide if we have no conflict or knowledge conflict or the ultimate form, contradiction. On the other side, when the situation dynamics is investigated, pretty often we turn to ontology corrections due to its incompleteness or incorrectness. In this situation the main conclusion for us is the following. We need use metaknowledge and dynamic ontologies to cope with conflict or contradiction identification. The conflict identification is almost always much more complicated than the contradiction case.

The ontology-based contradiction identification is followed by its resolution [8]. The proposed resolution methods are effective applications of ideas from nonclassical logics and they are one of base parts since may decades of the presented research in analogy inference machines, case-based methods, data mining, etc. Contradiction resolution depends on the situation and types of contradiction sides. Our research [8] revealed five main groups of resolution scenarios. Currently we make investigations to elaborate new contradiction resolution scenarios. The research shows that automatic contradiction resolution processes may stay active constantly using free computer resources; in other situations they may be directly activated by user. In the first case the knowledge and data bases will be constantly improved by continuous elimination of incorrect information or by improving the existing knowledge as a result of revealing and resolving contradictions. As a result our contradiction resolution methods have been upgraded to a machine learning method i.e. learning without teacher which is rather effective in case of IDS.

Two contemporary concepts may be shown how to make machine self-improvement leading to self-learning. The first one is based on the usage of artificial neural networks, or other heuristic methods. Those methods show low learning rate and high design costs. On contrary we offer machine self-improvement via contradiction or knowledge conflict resolution. The knowledge base is improving after every such resolution process. After the resolution, the *invariant* part of knowledge or method remains that makes it stronger and more flexible. This self-improvement needs only one time-consuming resource: juxtapositions between different groups of knowledge. It needs the human help only in some complex situations. The considered machine learning is an evolutionary process [12] and it gives better results if the intermediate solutions, *hypotheses* are tested in different models [13]. The system has many resources to constantly resolve the contradictions when no goal is given or in parallel to main jobs. We can't escape from heuristics but they are passed to the decision maker via productive human-machine interactions thus making the system alone more effective and less complex. Some part of heuristics is hidden in ontologies driving the process of learning. Most of computational discovery/data mining methods are data-driven. The considered research is more ontology-driven than data-driven but it belongs to the same group of methods. The below presented methods allow us to use not only statistical methods but also other knowledge acquisition methods for knowledge discovery.

This type of machine learning is novel and original in both theoretical and applied aspects.

3. Method Interactions under SMM Synthetic Metamethod Control

The described below methods interact under the common control of a new type of a synthetic metamethod (***SMM***). The considered metamethod avoids or *defeats* crossovers, phenotypes, mutations, or other elements from traditional evolutionary computation [11, 12]. Below we choose the formal description that is complemented with explanations in an analogous manner as the way to reduce the extra descriptions, because the general scheme of the chosen strategy is rather voluminous. *SSM* swallows and controls the following methods:

I. **INCONSISTENCY**: contradictions detection and resolution method;

II. **CROSSWORD** method;

III. **FUNNEL** method;

IV. **CALEIDOSCOPE** method.

A. The CROSSWORD Method

Let somebody tries to solve a problem with a complex sentence of 200+ letters with vague for the reader explanations. Let the unknown sentence be horizontally located. The reader can't solve the problem in an arbitrary manner, because the number of combinations is increased exponentially. Now it is convenient to **facilitate** the solution by linking the well known to the reader information with the complex one from the same model. The reader tries to find vertical words that he is conscious about like the place of KDS'1993 – Interhotel Sandanski... The more the crosspoints are, the easier is the solution of the horizontal sentence. The approach for the CROSSWORD is *even easier*. Here both the easy meanings and the difficult ones are from one domain, therefore there exists an additional help to find the final solution.

The difference of the CROSSWORD method from the usual crosswords is in its highly dimensional spaces and of course the analogy is rather far and is used only for the sake of brevity. Let G be a goal that must be solved, and it is decomposed into two types of subgoals: G_2 is deduced in the classical deductive manner using Modus Ponens, and G_1 is explored in the area defined by the constraints V_1-V_2-V_3 in fig. 5.

The constraints V_j are not necessarily linear. Nonlinear V_j are depicted in fig. 5. Let all the constraints are of different types. Denote V_1 is a curve dividing two groups: knowledge inconsistent with G_1 is located above V_1 and consistent knowledge is below the curve. Let V_1 divides the knowledge having accordance to G_1 from knowledge conflicting the subgoal. In the end let V_3 is a linear constraint e.g. x>1997. The solution to the subgoal lays inside the area depicted in fig. 5 and the goal resolution complexity falls significantly.

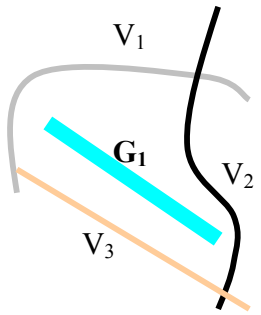


Figure 5. Example of nonlinear constraints

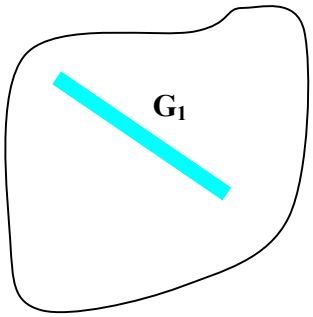


Figure 6. The goal is inside an ontology

Another situation reducing the resolution process is depicted in fig. 6 where the same subgoal G_1 is located inside and ontology which gives the search constraints. Sometimes the proof leading to the situation in fig. 6 is the proof *on contrary* when it is impossible the goal to be outside the considered ontology. Comparisons between two examples from fig. 5 and fig. 6 show that using ontologies to reduce the research area is more natural way and is much more effective than standard constraint satisfaction methodology.

Let subgoal G_1 is indeterminate or it is defined in a fuzzy way. Then the introduced algorithm is defined in the following way.

$$K_i \in K,\ i=1,2,\ldots n:\ G_1 \cap K_i \neq \varnothing;$$
$$L_j \in L,\ j=1,2,\ldots m:\ G_1 \cap L_j = \varnothing;$$
$$S=(G_1 \cap K_1), T=(G_1 \cap K_n); S \neq T; x_1,y_1,z_1 \in S;\ x_2,y_2,z_2 \in T; \quad (5)$$
$$\frac{x-x_1}{x_2-x_1}=\frac{y-y_1}{y_2-y_1}=\frac{z-z_1}{z_2-z_1}$$

where x_1, y_1, z_1 and x_2, y_2, z_2 are the coordinates of the respective boundary points from S and T from the set K whilst x, y and z are the coordinates of the points from the slice that tethers the explored area. In this way (by two sticking points) the goal search is restricted from an infinite space to a slice in the space. The introduced method is realized in an iterative manner: the goal place from (5) is replaced by K_i from the previous iteration and so on.

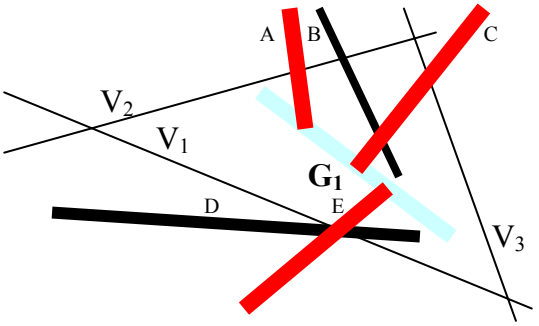


Figure 7. The CROSSWORD method goals and constraints

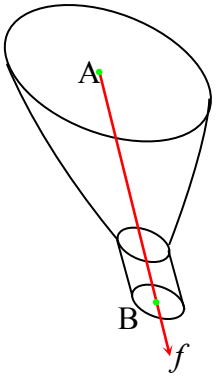


Figure 8. The FUNNEL method

Fig. 7 illustrates an example with three elements of K={A,B,C} where L={D,E} contains two elements. The example illustrates the benefit from the elements of L and from the spatial constraints V_u even in the case n>1. It is conspicuous that the direction of G_1 most often does not predetermine the integral decision and that the elements D,E and V_u decrease the number of the concurrent alternatives.

Different types of connections are depicted in fig. 7. The search restriction is done by V_1, V_2, and V_3 as considered above in fig. 5. The constraints B and D also restrict the search for G_1 but this restriction is dot-shape because B and D lay not in the search area bounded by V_j. On the other hand, those dots make tight fixation to G_1 , so the are denoted fixation constraints. In the end, A, C and E are resolution constraints because they intersect G_1 and give us parts of the solution to the problem.

B. The FUNNEL Method

We denote with $f(t_0)$ a fitness function in the point t_0. In the common case $f(t_0)$ may vary according to its environment – the position in the space and other impacts over the point. In this paper the function is linear and it does not change in the whole domain, $f=f(t_0)$. In this way $f(t_0)$ is reduced to a free vector f. Let $f(t_0)$ is one of the intermediate solutions to the goal when the process has reached up to t_0 . $f(t_0)$ points only to the *recommendable* direction for the evolution of the solution [12], so the movement in this direction shall be realized only if there are no other alternatives. Here we may use a 'gravity' analogy: it is too weak in case of e.g. jets but still it is enough strong not to be underestimated. $f(t_0)$ is combined with a system of spatial constraints in the following way:

$f(t_0)$ is the goal function;

$f_i(t_0)$ is a set of functions which affect t_0.

$$A\frac{d^n x}{d^n t}+B\frac{d^n y}{d^n t}+C\frac{d^n z}{d^n t}\le D \tag{6}$$

$$Ex+Fy+Gz\le H \tag{7}$$

where (6) is a system of non-linear constraints and (7) is a system of linear constraints. Then the direction of the solution in $f^*(t_0)$ is defined as a sum of the vectors multiplied by the respective coefficients k_i; the existing system of constraints is presented by (6) and (7).

$$f^*(t_0)=f(t_0)+\sum_i k_i f_i(t_0) \tag{8}$$

Let's assume you have a *plastic funnel*. If you fix it vertically above the ground, you can direct a stream of water or of vaporous drops etc. If you change the funnel direction, then the stream targeting will be hampered, if the

stream hasn't enough *inertion power*. Fixing the funnel horizontally makes it practically useless. Analogically in the evolutionary method the general direction in numerical models is determined likewise. In other words this is a movement along the predefined gradient of the information. Just like in the case of the physical example, there are lots of undirected hazardous steps towards conclusions and hypotheses in the beginning.

This paper offers the following modification of FUNNEL. Let k_i be not constants:

$$k_i(t_0) = \frac{k_i^0}{1 + D_0 - D} \tag{9}$$

where k_i^0 are the initial meanings coinciding with k_i from (8) and (t_0) are the respective coefficients in point t_0, D is the initial point in the investigated domain–a beginning of the solution and D_0 is an orthogonal projection of t_0 upon the straight line L parallel to *f* where $D \in L$. In this case moving away from the beginning D the solution depends more and more on the fitness function but the other external factors influence it less and less.

The FUNNEL method can be indirectly based on inconsistency tests with known information. The method may be used also in the other parts of *SMM*, e.g. in the CROSSWORD method it assists the determination of the direction of the explored goal. The graphical representation of the FUNNEL main idea is represented in Fig. 8.

It is a data driven method, so intruders haven't possibility to predict the results. The direction *f* from the figure is the goal, e.g. the fitness function from genetic algorithms. Unlike the other contemporary methods, the FUNNEL method gives the ISS freedom to choose and update the hierarchy of goals. In 'the loose part' A in Fig. 8, if a new goal appears and promises large gains, and if there is still a long way to resolve *f*, then ISS will try to reach the nearest goal, after that it will return to its way for *f*. The 'edge' constraints in FUNNEL are function of the following parameters: the 'stream inertia' of the intermediate solutions, 'gravity', etc. The next Sections show that INCONSISTENCY method also can be applied to define constraints in the FUNNEL method.

C. The CALEIDOSCOPE Method

The CALEIDOSCOPE is the visualization method: it presents the current results or the solution to the security expert. Apart from other interfaces, here some cognitive elements have been applied that help the user make conclusions using notions still unknown to the machine: 'beauty', 'useful', etc. Here the system role is mainly to inspire the decision making imagination and to give him the interesting results: repetitive patterns, etc. Many of the discribed methods already contain enough visualization elements, in these cases the CALEIDOSCOPE method makes only graphic interpretations of results. In other cases it should make an optimal rotation of the pattern or show intersections of pattern or make other processing helping the user make the decision in best comfort conditions.

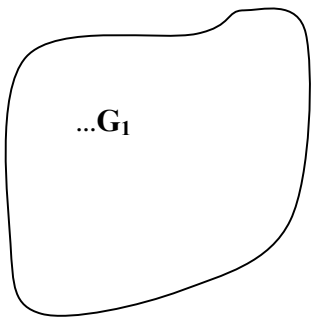


Figure 9. CROSSWORD -1: location of the goal

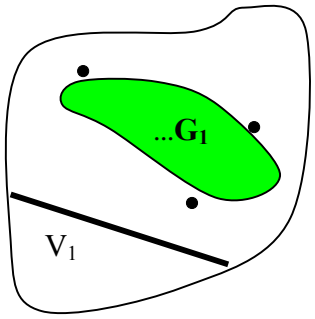


Figure 10. CROSSWORD -2. Two different constraint types

Fig 9 shows an example when the decision to the goal is located in the depicted ontology area, and all the other domain knowledge may be considered only if has some relation to the ontology. Let restriction constraint V_1 from fig. 10 is found e.g. 'show only new results', and three fixation constraints are found: the intersection of the curves with the ontology field is represented as three dots. Both two types of constraints make rough solutions thus helping to restrict the search area and make the method complexity better.

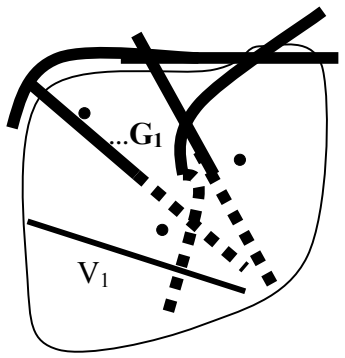


Figure 11. CROSSWORD -3

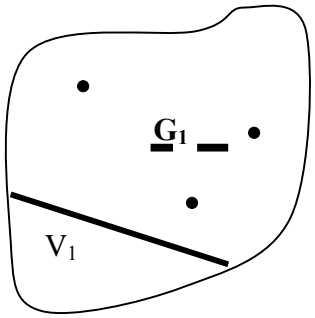


Figure 12. CROSSWORD -4

Fig. 11 shows same constraints and three resolution constraints making intersections with the desired goal G1. A part of other constraints helping define the three resolution constraints is depicted. In Fig. 12 is shown that the two right intersection parts are joined, and the left part is enlarged using knowledge modelling, binding and logic methods. Thus a big part of the goal is known and the security administrator will make correct conclusions.

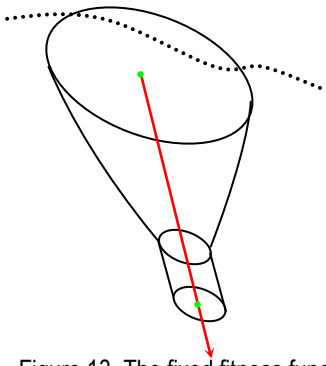


Figure 13. The fixed fitness function

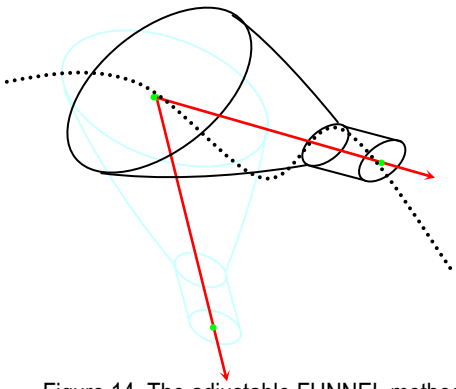


Figure 14. The adjustable FUNNEL method

The visualization of the FUNNEL method results is considered in Fig. 13. Let the solution to the problem, dotted line in fig. 13, has a 'strong inertia' thus leaving the desired area. The interpretation shows that the fitness function in this example should be shifted as shown in fig. 14, and then the solutions will go the desired direction. We use only a set of static pictures but it is obvious that multimedia and visualization of dynamic processes will make greater effect. The hope is to realize it in ongoing projects.

D. Interactions

Briefly, the synthetic control by SMM means that the overall result is defined 'by the design', by interactions between the methods much more than by the results given by each method itself.

Four methods are discussed in this section, INCONSISTENCY, CROSSWORD, FUNNEL, and CALEIDOSCOPE. There are much more methods under the SMM control: induction, juxtapositions etc. Not everything touching the problem is described to the sake of clarity and brevity. In general all the methods are collaborative as shown

above: FUNNEL-INCONSISTENCY-CALEIDOSCOPE in fig. 14 or CROSSWORD-CALEIDOSCOPE in fig. 11… On the other hand, all the methods are competitive where 'the fittest survives' principle means the following. As described, many processes should be executed in parallel but the computer resources are reserved for the high priority methods. The lowest priority is the constant inconsistence test, it runs if some free resources. The highest priority are user modeling, intruder modeling and expert- or user-ordered goals. Methods that brought many successful results in the past get higher priority. The security administrator may change the set of priorities.

Query processing, statistical inference and other knowledge discovery technologies may collaborate with the presented methods but they are included under the same SMM control. As stated above, our goal isn't to make a method that will substitute the best contemporary methods but SMM is a good addition to them. The wide part of the funnel in fig. 14 shows that the resolution process may start using statistics in the lack of knowledge and then go the desired direction when statistical methods are shifted by other knowledge acquisition methods. This important part of SMM is described in [12].

4. Realisations

The presented system source codes are written in different languages: C++, VB, OWL and Prolog. Many of the described procedures rely on the usage of different models/ ontologies in addition to the domain knowledge thus the latter are metaknowledge forms. In knowledge-poor environment the human-machine interactions have a great role, and the metaknowledge helps make the dialog more effective and less boring to the human. The dialog forms are divided in 5 categories from 1='informative' to 5='silent' system. Knowledge and metaknowledge fusion is always documented: where the knowledge comes from, etc. This is our main principle: every knowledge is useful and if the system is well organized, it will help us resolve some difficult situations.

We rely on nonsymmetric reply 'surprise and win', on the usage of unknown codes in combination with well known methods, and on the high speed of automatic reply in some simple cases e.g. to halt the network connection when the attack is detected. If any part of ISS is infected or changed aiming at reverse engineering or other goals, then the system will automatically erase itself and in some evident cracking cases a harmful reply will follow. The above represented models of users and environment are used in the case. Therefore different SMM realizations are not named IDS but ISS because they include some limited automatic reply to illegal activities.

The success of the presented applications is hidden in a rather simple realization of the presented methods. We tried to make complex applications using reasoning by analogy, machine learning or statistical data mining methods but in this case the complexity of SMM is greater than NP-hard.

5. Conclusions and Future Work

The main conclusion here is that many processes concerning human-machine creation are onology-based. Our additional purpose is to show that when the machine helps to resolve the problem using its strongest features/formal part and the heuristic part/emotions/notions like simple, beautiful, interesting are left to the decision maker, then the human-centered methods are rather effective and simple.

There exists no free cheese… To make an advanced system we should define and use many labor-consuming ontologies. In perspective we hope use machine-learning or other knowledge acquisition methods to construct ontologies. In parallel we use the considered methods in older projects [14].

Bibliography

[1] M. Miller. Absolute PC Security and Privacy. SYBEX Inc., CA, 2002.

[2] D. Song, M. Heywood, A. Zincir-Heywood. Training Genetic Programming on Half a Million Patterns: An Example From Anomaly Detection, IEEE Trans./Evolutionary Computation, no. 3, pp. 225-239, 2005.

[3] H. Kyburg, *Probability and Inductive Logic*, Progress, Moscow, 1978.

[4] The Handbook of Data Mining, N. Ye (Ed.), Lawrence Erlbaum Associates, NJ, 2003.

[5] S. Denchev and D. Hristozov. Uncertainty, Complexity and Information: Analysis and Development in Fuzzy Information Environment. Zahari Stoyanov, Sofia, 2004.

[6] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, Prentice Hall, NJ, 1997.

[7] V. Jotsov. "Knowledge discovery and data mining in number theory: some models and proofs," Proc. Methods and Algorithms for Distributed Information Systems Design. Institute for Information Transmission Problems of RAS, Moscow, pp.197-218, 1997.

[8] V. Zgurev and V. Jotsov, "An approach for resolving contradictions," *J. Controlling Systems and Machines* Vol. 7-8 , pp. 48-59, 1992.

[9] A. Arruda, "A survey on paraconsistent logic," *in Math. Logic in Latin America*, A. Arruda, C. Chiaqui, N. Da Costa, Eds. North-Holland, Berlin NY, pp. 1-41, 1982.

[10] D. E. Goldberg, The Design of Innovation Lessons from and for Competent Genetic Algorithms, Kluwer, NY etc., 2002

[11] A. Goel, "Design, analogy and creativity," *IEEE Expert/Intelligent Systems and Their Applications*, vol. 12, no. 3, May 1997.

[12] V. Jotsov. "Evolutionary parallels," *Proc. First Int. IEEE Symp. 'Intelligent Systems'*, T. Samad and V. Sgurev (Eds.), Varna, Bulgaria, vol. 1, pp. 194-201, 2002.

[13] V. Jotsov. "Knowledge acquisition during the integer models investigation," *Proc. XXXV Int.Conf. "Communication, Electronic and Computer Systems"*, Technical University of Sofia, pp. 125-130, 2000.

[14] V. Jotsov, V. Sgurev. "An investigation on software defence methods against an illegal copying," *Proc. IV Int. Sci. Conf. 'Internet - an environment for new technologies'*, vol. 7, V. Tarnovo University 'St. St. Kiril and Metodius', pp. 11-16, 2001.

Author's Information

V.S.Jotsov (В.С. Йоцов): *State Institute of Library Studies and Information Technologies;*

Intsitute of Information Technologies of the Bulgarian Academy of Sciences.

P.O.Box 161, Sofia 1113, Bulgaria

RULE-MINING: ПОДХОД К АВТОМАТИЗИРОВАННОМУ ИЗВЛЕЧЕНИЮ ОНТОЛОГИЙ

Роман Гадиатулин, Светлана Чуприна

Аннотация*: В статье описывается подход к реализации автоматизированного извлечения онтологий из продукционных баз знаний, вопросы организации извлечения, хранения и использования онтологий для широкого круга проблемных областей. Разработанная в рамках предлагаемого подхода оболочка OntoMaker 2.0, наряду с возможностями автоматизированного извлечения знаний предоставляет визуальные средства для редактирования и построения онтологий, а также широкие возможности расширения и настройки.*

Keywords*: извлечение знаний, автоматическая каталогизация и классификация, онтологический подход, проектирование и разработка интеллектуальных информационных систем.*

ACM Classification Keywords*: I.2 Artificial Intelligence: I.2.7 Natural Language Processing – Text analysis; D.2 Software Engineering: D.2.2 Design Tools and Techniques – Computer-aided software engineering (CASE).*

Введение

Современное состояние дел в отдельных областях компьютерных наук диктует необходимость привлечения методов инженерии знаний для решения широкого класса практических задач. Ярким примером тому является инициатива Semantic Web, основная цель которой – наделить огромные массивы данных, опубликованных в сети Internet большей осмысленностью, повысить удобство работы с этой информацией. Одним из главных достижений проекта Semantic Web стала разработка стандарта описания онтологий – OWL (Ontology Web Language), благодаря чему множество инженеров по знаниям, программистов и экспертов получили возможность использовать общие правила представления, хранения и обработки онтологий.

Существуют различные толкования самого понятия онтологии [Гаврилова,2000]. В данной работе под онтологией понимается структурная спецификация некоторой предметной области, ее концептуальное описание в виде формализованного представления, которое включает словарь терминов предметной области и логические выражения, описывающее взаимосвязи этих понятий. Таким образом, онтология некоторой предметной области представляет собой тезаурус понятий этой предметной области, обеспечивающий возможность толкования терминов предметной области посредством интерпретации таких типов парадигматических отношений как «часть-целое», «класс-подкласс» и некоторых видов ассоциативных связей.

На волне интереса к онтологиям были созданы инструментальные средства и механизмы, специально ориентированные на широкое применение онтологий в задачах интеллектуального поиска, классификации, выявление несогласованности в данных, моделирования поведения интеллектуальных агентов. Однако даже наличие хорошего инструментального окружения не снимает проблем, связанных с трудностью проектирования и построения самих онтологий, а автоматизация процесса извлечения онтологий, как и в целом, задача извлечения знаний, и по настоящее время не имеют своего эффективного решения. Тем ценнее становятся уже разработанные онтологии и опыт их использования для решения широкого круга задач.

Кроме того, к настоящему времени накоплен большой объем баз знаний (БЗ), созданных не только в рамках различных инструментальных сред, но и на основе различных парадигм представления знаний (продукционной, логической, фреймовой, на семантических сетях). В процессе создания современных интеллектуальных информационных систем зачастую требуется интеграция знаний из разнородных источников и, как следствие, эффективное решение задач, связанных с тиражированием знаний. По-прежнему не имеет своего удовлетворительного решения проблема автоматизации процесса выбора адекватного специфике конкретной проблемной области и принятого в ней стиля рассуждения экспертов средства представления знаний. Поэтому и по сей день актуальны исследования, направленные на разработку такого подхода к представлению и тиражированию знаний, который с одной стороны позволял бы наиболее адекватно учитывать специфику проблемной области, а с другой – представлять и использовать знания в некотором унифицированном виде.

В данной статье мы не касаемся вопросов, связанных с решением указанных проблем на основе применения современных мультиагентных технологий. Нам представляется весьма актуальным применение онтологий для решения перечисленных выше проблем в рамках интеллектуальных информационных систем, основанных на использовании баз знаний традиционных экспертных систем (ЭС), которые по-прежнему остаются одними из самых распространенных приложений искусственного интеллекта на практике. Предлагаемый нами в рамках проекта XG# подход к извлечению онтологий проблемных областей предусматривает автоматизацию процесса анализа уже существующих продукционных баз знаний (rule-mining) и выявления в них скрытых зависимостей, которые могут быть явно представлены в виде основных парадигматических отношений. Построенная таким образом онтология в значительной мере способствует обнаружению несогласованностей и противоречий в исходной БЗ, а также облегчает ее сопровождение и отладку. Кроме того, благодаря поддержке стандарта

OWL, обеспечивается возможность тиражирования знаний между различными когнитивными системами, что автоматизирует процесс создания и пополнения баз знаний этих систем.

Проект XG#

XG# – это инструментальная среда разработчика оболочек экспертных систем с интегрированным представления знаний, основанная на декларативном описании грамматики языка представления знаний, учитывающего специфику конкретной проблемной области [Чуприна,2003-2006], [Гадиатулин,2006]. Описание грамматики конкретного языка представления знаний в среде специально разработанного визуального инспектора грамматик транслируется во внутреннее представление системы, на основе которого генерируются все основные компоненты оболочки ЭС (компоненты приобретения знаний, логического вывода, объяснения).

В основе предлагаемого подхода – активное использование метазнаний не только для описания синтаксиса и семантики языка представления знаний, но и для формирования онтологий, описывающих основные виды взаимосвязей между понятиями проблемной области, а также модель пользователя системы. Однако, как уже отмечалось во введении, проектирование и разработка онтологий, то есть онтологический инжиниринг, не является тривиальной задачей. Он требует от разработчиков профессионального владения технологиями инженерии знаний — от методов извлечения знаний до их структурирования и формализации [Гаврилова,2000].

Перечислим несколько важных, на наш взгляд, задач, решение которых можно осуществить при помощи привлечения онтологий за счет явного представления и учета основных парадигматических отношений между понятиями проблемной области:

- упрощение отладки и модификации базы знаний ЭС за счет визуализации связей между понятиями, представленными в БЗ системы;
- обеспечение дополнительных знаний для автоматической генерации объяснений хода логического вывода;
- тиражирование знаний посредством публикации онтологий, полученных с помощью автоматизации процесса извлечения скрытых знаний из баз знаний уже существующих ЭС или созданных экспертом «с нуля» в среде визуального редактора онтологий;
- ускорение процесса разработки ЭС за счет автоматизации этапа первоначального заполнения БЗ на основе существующих в данной проблемной области онтологий;
- трансформация и приведение к единому виду знаний из БЗ различных ЭС.

Методика rule-mining

В проблематике искусственного интеллекта большой популярностью в настоящее время пользуются подходы, основанные на автоматизации извлечения знаний из данных, текстов и веб-ресурсов (data-mining, text-mining, web-mining). В рамках проекта XG# мы исследовали возможность автоматизации извлечения знаний более высокого уровня из предметных баз знаний продукционных ЭС. Такой подход можно условно назвать rule-mining. Исходными данными для rule-mining служит текст базы знаний, из которого автоматически извлекаются закономерности, которые в явном виде представляются в среде визуального редактора онтологии для целей дальнейшего анализа и редактирования пользователем, что в значительной мере упрощает процесс построения онтологии проблемной области. В результате реинжиниринга из БЗ автоматически извлекаются дерево целей и описание доменов значений понятий, представленных в ЭС. Процесс извлечения онтологии схематично представлен на рис.1.

Построитель онтологии анализирует возможность наличия взаимосвязей между понятиями БЗ на основе некоторых эвристик. Например, в результате анализа фактов, содержащихся в посылке и заключении правила, система может сделать заключение о наличии между ними связи типа «подкласс-класс». Далее

автоматически на основе метазнаний генерируются наборы вопросов пользователю-эксперту с целью уточнения семантики выявленных взаимосвязей, анализируются ответы пользователя и строится соответствующая онтология. Результирующая онтология таким образом в явном виде отражает различные виды взаимосвязей (например, «класс-подкласс», «часть-целое») понятий проблемной области, которые неявно содержатся в фактах и правилах исходной БЗ предметного уровня. Далее онтология автоматически сохраняется в виде описания классов, экземпляров и свойств в нотации OWL и может быть с успехом использоваться для решения перечисленных выше задач.

Результирующая онтология, возможно, является неполной и неточной, однако эксперт имеет мощное средство для ее визуального редактирования и пополнения.

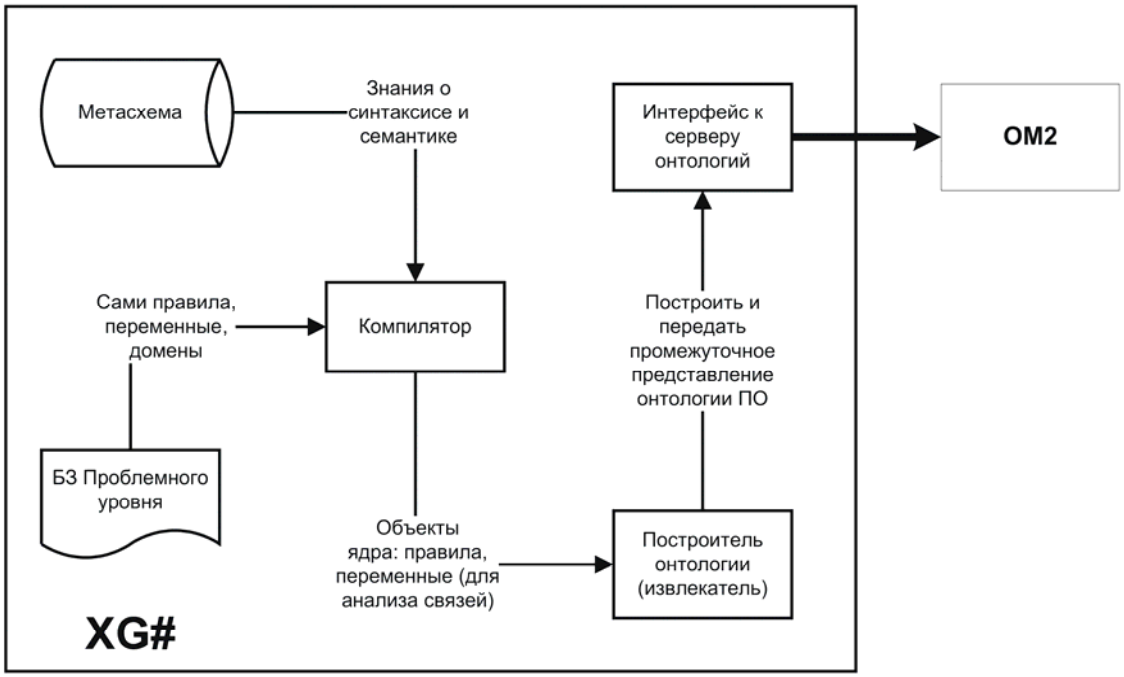


Рис. 1. Процесс извлечения онтологии

Визуальный инструментарий для работы с OWL онтологиями OntoMaker 2.0 (OM2)

На сегодняшний день для большинства инструментальных средств построения онтологий характерно следующее. Во-первых, большая часть подобных систем хоть и имеет визуальную составляющую, однако некоторые конструкции приходится набирать вручную, что повышает уровень требований к разработчику онтологий – перед тем, как приступать непосредственно к своей работе, эксперт вынужден тратить время на изучение языка представления знаний. Во-вторых, часть инструментальных средств реализуют определенную функциональность для выполнения запросов к онтологиям, но, к сожалению, не имеют унифицированного интерфейса для формирования и выполнения запросов из внешних приложений. В-третьих, практически нет свободно распространяемых и ориентированных на конечного пользователя редакторов онтологий, что, естественно, замедляет развитие всего направления онтологического инжиниринга.

При разработке инструментальной среды разработчика онтологий OM2 мы постарались нивелировать минусы аналогов и перенять их достоинства. В OM2 любой объект онтологии имеет графическое представление (не только классы и индивиды, но и свойства, связи и др.). OM2 – независимое приложение, которое способно выступать в качестве сервера онтологий. В настоящее время редактор полностью поддерживает конструкции диалекта Lite языка описания онтологий OWL, ведется работа по расширению ее возможностей до диалекта DL. На рис. 2 приведена архитектура OM2.

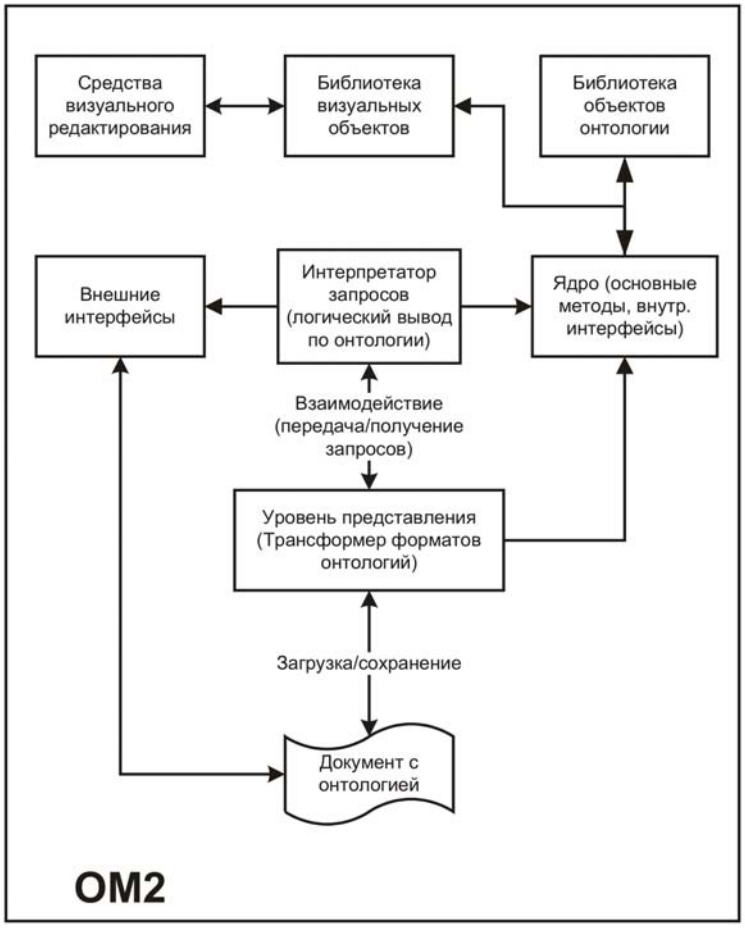


Рис. 2. Архитектура OM2

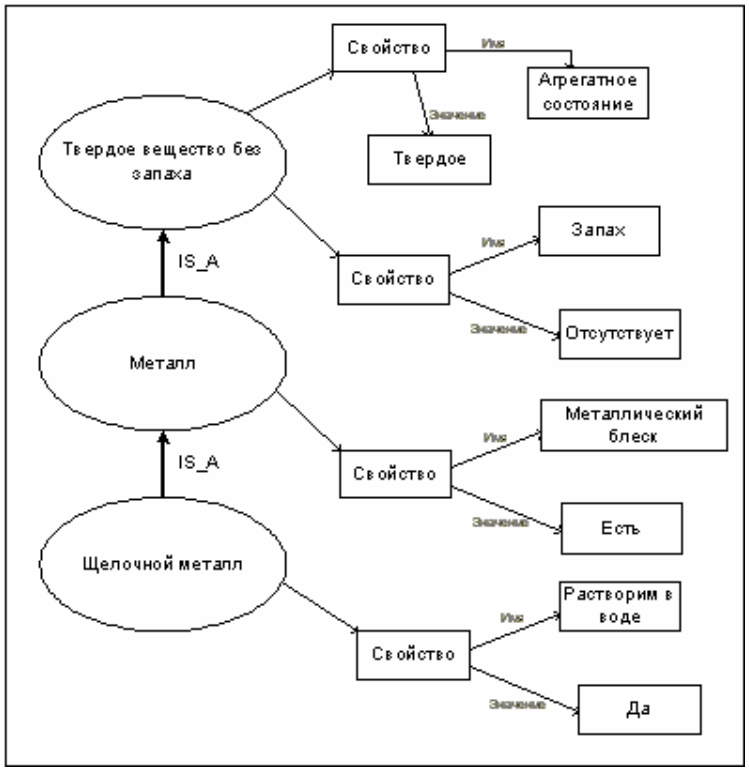


Рис. 3. Фрагмент онтологии базы знаний демонстрационной ЭС по определению химических веществ

В заключение отметим, что описанный подход прошел апробацию на нескольких международных конференциях (см. [Чуприна,2004-2006], [Гадиатулин,2006]). Обнадеживающие результаты были получены при использовании OntoMaker версии 1.0 для автоматизированного извлечения онтологий из БЗ экспертной системы по определению химических веществ, разработанной в Пермском государственном университете, и ряда других ЭС, находящихся на стадии исследовательского прототипа, что подтверждает жизнеспособность предлагаемого подхода.

На рис.3 приведен фрагмент онтологии, автоматически построенной в разрабатываемой нами инструментальной среде на основе анализа базы знаний демонстрационной ЭС по определению химических веществ.

Результаты

Область применения онтологий, построенных вручную с помощью визуального инструментария OM2, внешних онтологий, опубликованных в Web, или онтологий, извлеченных посредством rule-mining, очень широка: от обнаружения и исправления ошибок при вводе (модификации) знаний до автоматической генерации на естественном языке осмысленных объяснений хода логического вывода ЭС. Сам процесс автоматизированного построения онтологий способствует более глубокому анализу проблемной области и упрощению отладки ЭС. Создав онтологию, эксперт облегчает работу себе, пользователю и, возможно, другим разработчикам экспертных систем в данной проблемной области. Само по себе существование онтологии, естественно, не решает описанных в данной статье проблем. Для этого онтология должна быть интегрирована в систему таким образом, чтобы было обеспечено ее активное и эффективное использование всеми компонентами экспертной системы, а также возможность ее дальнейшего наращивания и пополнения, в том числе посредством тиражирования знаний из внешних источников.

Построение онтологии не только цель, но и средство, облегчающее разработку ЭС:

Во-первых, при автоматическом формировании дерева целей упрощается отладка, появляется инструмент для контроля непротиворечивости знаний, наличия циклических ссылок, изолированных веток вывода и т.д. Эксперт и инженер по знаниям получают возможность проанализировать каузальные зависимости, сравнить выявленные взаимосвязи между понятиями БЗ экспертной системы с реально существующими в соответствующей проблемной области. Автоматически построенное дерево целей является хорошим дополнением к традиционным средствам отладки ЭС.

Во-вторых, на основе автоматизированного разбора БЗ, выявленные и представленные в явном виде знания о каузальных зависимостях облегчают реинжиниринг базы знаний. Занесение и удаление правил, создание и удаление переменных становится легче контролировать: все изменения можно отслеживать по дереву целей и выдавать соответствующие предупреждения. Аналогичным образом упрощается задача миграции базы знаний в другую оболочку (например: из Guru в XG#).

В-третьих, возникает возможность разделения онтологий между экспертными системами для одной проблемной области, тиражирования наиболее удачных, общепризнанных онтологий, что влечет сокращение общего срока разработки ЭС или другой системы, основанной на тиражировании и использовании знаний.

Библиографический список

[Гаврилова,2000] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем: Учебник для вузов. — СПб: «Питер», 2000.

[Чуприна,2003] Жигалов А.В., Никулин М.Б., Чуприна С.И. Оболочка экспертных систем XG#: концепция построения и реализация // Математика программных систем: межвуз. сб. научн. тр. / Перм. ун-т. Пермь, 2003. С. 97-106.

[Чуприна,2004] Chuprina S., Lanin V., Borisova D., Khaeva S. Internet Intelligent Search System SmartFinder // Proc. of the European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology. Knowledge-Based

Media Analysis for Self-Adaptive and Agile Multimedia Technology/ The Royal Statistical Society, November 25–26, 2004, London, U.K. P. 151–156.

[Чуприна,2005] Chuprina S., Nikulin M. XG#: the Toolkit for Integrated Knowledge Representation Expert System Shells Development// Advances in Current Natural Sciences. Vol.5, 2005. Proc. of XXXII International Conference IT+S&E'2005. Ukraine, Crimea, Yalta-Gurzuff, 2005. P. 105.

[Чуприна,Гадиатулин,2006] Чуприна С.И., Гадиатулин Р.А. XG# 1.0: Использование метазнаний для интеграции различных баз знаний //Актуальные проблемы математики, механики, информатики: материалы Международной научно-методической конференции, посвященной 90-летию высшего математического образования на Урале / Перм. гос. ун-т. Под ред. Лядовой Л.Н., Яковлева В.И., Ясницкого Л.Н. – Пермь, 2006. С. 104-105.

[Чуприна,2006] Чуприна С. И. Интеллектуальная поисковая система SmartFinder: подход на основе онтологий // «Вестник Пермского университета», серия «Математика. Механика. Информатика», выпуск 4(4) / Перм. гос. ун-т. Пермь. 2006. С.118-122

[Чуприна,Лядова,Ланин,2006] Чуприна С.И., Лядова Л.Н., Ланин В.В. Система интеллектуального поиска и автоматической каталогизации документов на основе онтологий // Proceedings of the XII-th International Conference “Knowledge-Dialogue-Solution” (KDS 2006), June 20-25, 2006, Varna (Bulgaria).- Sofia: FOI-COMMERCE, 2006, С. 139-144.

[Гадиатулин,2006] Гадиатулин Р.А. Чуприна С.И. XG# 1.0 Реализация подхода к автоматическому построению онтологии из текстов баз знаний. Научно-технический, рецензируемый журнал общественного объединения Белорусской инженерной академии 1(21)/1 '2006 с. 39-42.

Информация об авторах

Роман Гадиатулин *– Пермский государственный университет, студент магистратуры кафедры математического обеспечения вычислительных систем; Россия, г. Пермь, 614990, ул. Букирева, д. 15; e-mail: gadiatulin@ramlber.ru*

Светлана Чуприна *– Пермский государственный университет, доцент кафедры математического обеспечения вычислительных систем; Россия, г. Пермь, 614990, ул. Букирева, д. 15; e-mail: chuprinas@inbox.ru*

ОНТОЛОГИЧЕСКИЙ АНАЛИЗ WEB-СЕРВИСОВ В ИНТЕЛЛЕКТУАЛЬНЫХ СЕТЯХ

Анатолий Гладун, Юлия Рогушина, Виктор Штонда

Аннотация*: Определены подходы к анализу семантики Web-сервисов на основе их онтологий, ориентированные на автоматизацию их поиска и компоновки в интеллектуальных сетях. Проанализированы средства создания онтологических описаний Web-сервисов. Предложены алгоритмы формирования и усовершенствования таких онтологий, а также методы их сравнения.*

Ключевые слова*: Web-сервис, онтология, тезаурус, поиск информации.*

ACM Classification Keywords*: I.2.4 Knowledge Representation Formalisms and Methods*

Введение

Центральным элементом зарождающейся сервис-ориентированной архитектуры (SOA) является способность разрабатывать новые приложения посредством композиции функциональных возможностей

предприятия, представленных в форме сервисов - в пределах одной организации или же среди множества предприятий. Семантические описания сервисов, включая аннотации функциональных и не функциональных атрибутов, направлены на автоматизацию этого процесса и создание более качественных решений. Необходимо описать возможности сервисов на однозначно интерпретируемом, поддающемся компьютерной обработке языке и улучшают качество и устойчивость существующих задач, таких, как Web-сервисы и вызовы процедур. Одна из целей проекта Semantic Web – создать средства, позволяющие пользователям выбирать, использовать, компоновать и отслеживать такие Web-сервисы автоматически. Онтологии, в общем виде определяемые как совместно используемые формальные концепции конкретных предметных областей (ПрО), в состоянии сыграть критически важную роль в организации обработки знаний на базе Web, их совместного использования и обмена ими между приложениями.

Концепция Web-сервисов

Сервис-ориентированная архитектура (СОА) – это парадигма проектирования, разработки и управления функциональных модулей (сервисов), каждый из которых доступен через сеть и способен выполнять определенные действия. СОА создает коммуникационную среду для модулей, реализующих прикладную бизнес-логику. Под сервисом понимают некоторый Web-сайт, который предоставляет не просто статичную информацию, но также позволяет выполнять некоторые действия или изменять мир, например, продавать какие-либо продукту или управлять физическим устройством [1]. Web-сервисы обеспечивают высокоуровневые абстракции для того, чтобы организовать применение их в крупномасштабных, открытых средах. Web-сервис – это автономное приложения, предоставляющее средства доступа к информации внешним клиентам через набор предоставляемых им услуг. В [2] Web-сервис определяется как самоуправляемый модуль, способный обрабатывать данные и оперировать ими; он имеет возможность взаимодействовать с окружающей средой с помощью сообщений. В одном модуле соединены функции хранения данных и их обработки; сервис создается и внедряется независимо от других, поэтому может быть изменен без влияния на партнерские сервисы; сервис имеет четко определенные границы и представлен извне только через сообщения. Иначе говоря, сервис может быть определен как компонент IT, который обеспечивает выполнение взаимозависимых функций и общается с внешней инфраструктурой с помощью обмена сообщениями. Основная задача заключается в группировании функций и обеспечении их механизмами обмена сообщениями.

Сервисы могут быть простыми, или "примитивными", в том смысле, что они активизируют только какую-то одну доступную через Интернет программу, сенсор или устройство, не используя другие Web-сервисы, и поэтому взаимодействие между пользователем и сервисом, кроме простого вызова, отсутствует. Например, сервис, который возвращает почтовый код или широту и долготу, по указанному адресу. Они могут быть и составными, т.е. состоящими из нескольких примитивных. Такие сервисы обычно требуют взаимодействия или диалога между пользователем и сервисами, т.е. пользователь может осуществлять выбор или накладывать определенные условия. Пример – покупка книги через www.amazon.com: пользователь ищет книги по различным критериям, а затем решает, покупать ли их и затем предоставляет кредитную карту или почтовый адрес.

Web-сервисы базируются на трех основных Web-стандартах: SOAP (Simple Object Access Protocol) — протоколе для посылки сообщений по протоколу HTTP и другим Интернет-протоколам; WSDL (Web

Services Description Language) – языке для описания программных интерфейсов Web-сервисов; UDDI (Universal Description, Discovery and Integration) – стандарте индексации Web-сервисов

Постановка задачи

Цель работы состоит в том, чтобы, проанализировав принципы и технологию сервис-ориентированных вычислений, области применения Web-сервисов, предложить методы анализа семантики Web-сервисов, направленные на автоматизацию их компоновки и основанные на онтологическом анализе. Кроме описания семантики самого сервиса, важно соотнести его с определенной ПрО, в терминах которой и определяются его возможности и ограничения. Для этого необходимо формализовать знания и представления разработчиков сервиса и представить их в форме онтологии, пригодной для автоматической обработки и опубликовать их, сделав доступными для потенциальных пользователей сервиса. Это вызывает потребность в методах создания, усовершенствования и анализа онтологий ПрО.

Компоновка Web-сервисов

Возможность компоновки (composability) часто рассматривают как одно из основных преимуществ Web-сервисов. Компоновка Web-сервиса состоит из нахождения набора атомарных сервисов, необходимых для реализации запроса пользователя, и определение порядка их выполнения.

Компоновка Web-сервисов подобна проблеме планирования, которая исследовалась в искусственном интеллекте. Но классические планировщики непригодны для компоновки Web-сервисов, потому что предназначены для работы в статичном окружении (а среда Интернет динамична) и не имеют полной информации о системе, с которой работают (например, о возможностях различных сервисов).

Для автоматической компоновки программы должны уметь отбирать нужные им Web-сервисы и комбинировать их для достижения своих целей. Таким образом можно строить совершенно новые сервисы, комбинируя сервисы, уже имеющиеся в сети. Информация, содержащаяся в реестре UDDI, недостаточна для того, чтобы автоматически выполнить компоновку Web-сервисов, так как необходимы усилия человека для интерпретации семантики этих сервисов. Поэтому необходимо разрабатывать механизмы отображения семантики сервисов, запросов пользователей этих сервисов и их автоматизированного сопоставления с учетом специфики предметной области (ПрО), интересующей пользователя. Для композиции Web-сервисов в динамичной среды необходимо иметь больше синтаксической и статической метаинформации о них: Web-сервисы в гетерогенной среде часто оказываются недействительными, меняются их версии или они заменяются другими сервисами.

Интеллектуальные сети

Понятие интеллектуальных сетей (Intelligent Networks, или IN) возникло в связи с потребностью человечества в «умных» в сетях, которые могли бы сами подстраиваться под нужды пользователей, то есть разумно выполнять различные операции по хранению, обработке, передаче, систематизации информации, а также реализовать различные сервисы [3].

Интеллектуальные сети (IN) базируются на платформах традиционных телекоммуникационных сетей и включают интеллектуальную надстройку, обеспечивающую предоставление пользователям интеллектуальных сервисов различного назначения. В настоящее время телекоммуникационные и компьютерные приложения и услуги развиваются в направлении телематических услуг (мультилингвистические системы, интеллектуальные системы поиска информации, автоматизация бизнес-процессов, телемедицина, дистанционное обучение, системы обработки мультимедийной

информации и др.). Кроме того, сегодня наблюдается процесс интеграции различных сетей (мобильных и наземных, специализированных сетей Ad-hoc, VPN с сетями общего (публичного использования (Internet, Frame Relay и т.д.).

Интеллектуальная сеть - это вид глобальной сети для которой характерно:

- широкое использование различных методов обработки информации;
- эффективное использование ресурсов сети;
- модульность функций сети;
- интегрированные возможности разработки и внедрения услуг средствами модульных и многоцелевых сетевых функций;
- гибкое распределение сетевых функций по физическим элементам сети;
- возможность перемещения сетевых функций из одного физического элемента сети в другой;
- стандартизованное взаимодействие сетевых функций посредством независимых от услуг сетевых интерфейсов;
- возможность управления некоторыми атрибутами сервисов пользователями;
- стандартизованное управление логикой сервисов.

Набор услуг продолжает постоянно пополняться, однако он еще недостаточно широк и находится сегодня на стадии динамического развития. Основная проблема заключается в том, чтобы найти и скомпоновать уже существующие услуги таким образом, чтобы решит задачу, стоящую перед пользователем.

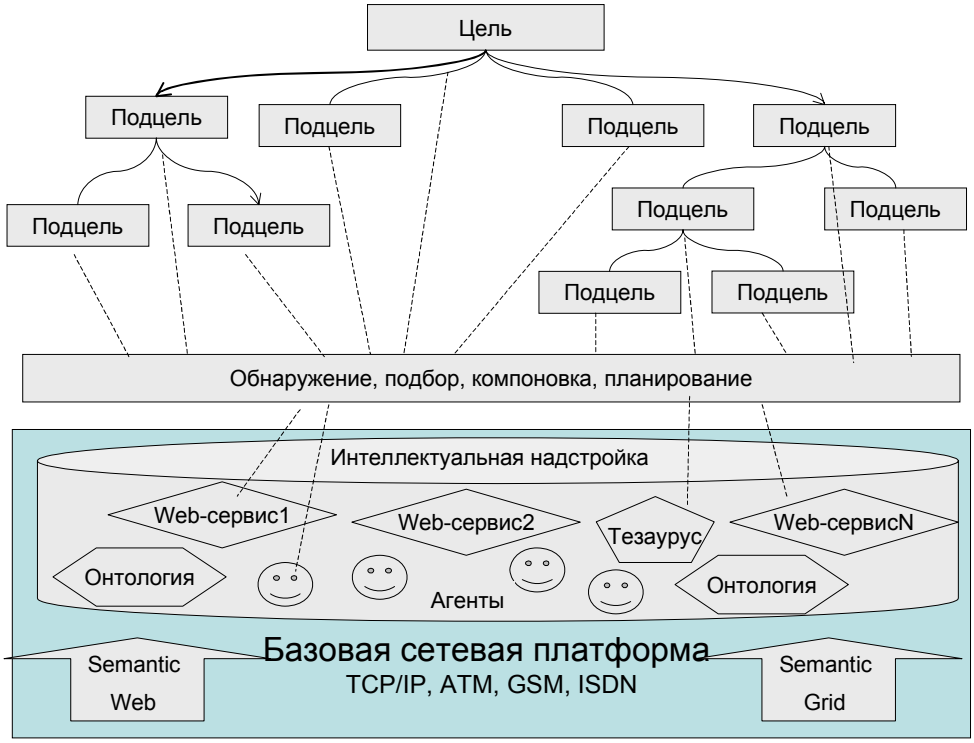


Рис. 1 Модель реализации интеллектуальных услуг на основе IN

В современных IN интеллектуальная надстройка использует онтологии, программные агенты и технологию Web-сервисов, позволяющие представлять и применять знания (рис.1). Так, целью проектов Semantic Web и Semantic Grid и является переход от традиционного Интернет к распределенной базе

знаний, предлагающей пользователям интеллектуальные услуги и обрабатывающей контент (как статичный, так и динамичный) на семантическом уровне.

Интеллектуальные Web-сервисы призваны кардинально улучшить взаимодействие людей и информационных систем, представить пользователям и корпорациям интеллектуальные услуги, обеспечить взаимное проникновение разных систем и процессов и обеспечивают новый этап в развитии современных интеллектуальных сетей.

Онтология OWL-S: семантическая разметка Web-сервисов

Чтобы использовать Web-сервис, программные агенты нуждаются в интерпретируемом компьютером определении сервиса и средств доступа к нему. Важной целью языков разметки проекта Semantic Web является установление структуры, в которой такие описания можно создавать и распространять. Web-сайты должны использовать набор базовых классов и свойств для декларирования и описания сервисов, и именно онтологические механизмы структурирования OWL обеспечивают подобную структуру.

OWL-S предназначен для того, чтобы помочь пользователям и поисковым агентам, обнаруживать, вызывать, компоновать и контролировать Web-сервисы [4]. Цель OWL-S – сделать возможным использование логического вывода для Web-сервисов, планирование компоновки Web-сервисов, автоматическое использование сервисов программными агентами.

OWL-S – это онтология сервисов, реализующая эту функциональность. Толчком к его разработке стал проект Semantic Web, который стремится обеспечить доступ к ИР Web не по ключевым словам, а по контенту. Важным результатом в этом направлении стала разработка языков разметки Web, таких как OWL и его предшественник DAML+OIL. Эти языки позволяют создавать онтологии любой ПрО и устанавливать связи с этими онтологиями Web-сайтов для их описания.

OWL-S обеспечивает декларативные описания свойств Web-сервиса и возможности, которые могут использоваться для автоматического обнаружения сервиса, декларативный API для автоматизированного выполнения Web-сервисов, которые являются необходимыми для Web-сервисов. Структурирование онтологии сервисов связано с тремя типами знаний о сервисах:

- Профиль сервиса – какую информацию сервис требует от пользователя или агента и что сервис ему предлагает: класс SERVICE представляет SERVICEPROFILE;
- Модель сервиса – как он работает: класс SERVICE описывается в SERVICEMODEL;
- Основание – как это использовать: класс SERVICE поддерживается в SERVICEGROUNDING.

Для каждого опубликованного сервиса должен существовать один экземпляр класса SERVICE.

Профиль сервиса строится на основе контента UDDI, описывающем свойства сервиса, необходимые для его автоматического обнаружения, такие, например, как предложение сервиса, его входы и выходы, предварительные условия и дополнительные действия. На основе профиля, который представляет информацию о провайдере, функциональных возможностях, и функциональных атрибутах сервиса, могут быть созданы описания и запросы сервиса. Профиль Web-сервиса дает описание его свойств: категорию сервиса (например, по классификации UNSPSC) и его качественную оценку (скорость, надежность и т.п.). OWL-S решает задачи четырех типов:

Автоматическое обнаружение Web-сервисов. Автоматическое обнаружение (discovery) Web-сервисов включает в себя автоматическое определение местонахождения Web-сервисов, которые предоставляют определенную услугу и удовлетворяют наложенным ограничениям. Например, пользователь хочет найти сервис по продаже авиабилетов между двумя определенными городами, принимающий его кредитную карту. Если сервис размечен при помощи OWL-S, то семантическая информация, необходимая для

нахождения сервиса, представлена на сайте Web-сервиса в форме, интерпретируемой компьютером, и тогда поисковая система, способная обрабатывать онтологическую информацию, может обнаружить такой сервис автоматически. Альтернативный подход заключается в том, что сервер может проактивно помещать в некоторые реестры сервисов OWL-S-описание предоставляемого им сервиса, а такие реестры могут быть найдены по соответствующему запросу. Таким образом, OWL-S должен обеспечивать декларативное представление свойств и возможностей сервиса, которые могут использоваться для автоматического обнаружения этого сервиса.

Автоматический вызов Web-сервиса. Автоматический вызов (invocation) Web-сервиса заключается в автоматическом выполнении идентифицированного ранее Web-сервиса компьютерной программой или агентом. Например, пользователь может запросить покупку с определенного сайта авиабилета на конкретный рейс. Если не использовать Web-сервис, то пользователь должен выполнить вручную набор действий: зайти на этот сайт, заполнить определенную форму и нажать на кнопку для выполнения сервиса или же отправить запрос с соответствующими параметрами непосредственно к сервису. Выполнение Web-сервиса можно рассматривать как последовательность вызовов функций. Разметка OWL-S Web-сервиса должна обеспечивать декларативный, интерпретируемый компьютером API для выполнения этих функциональных вызовов. Программный агент должен быть способен интерпретировать разметку, чтобы понять, какие нужны входные данные для вызова сервиса, какая информация будет возвращена и как выполнить сервис автоматически. Таким образом, OWL-S должен обеспечивать декларативные API для Web-сервисов, которые нужны для их автоматического выполнения.

1. **Автоматическая композиция и взаимодействие Web-сервисов.** Эта задача включает в себя автоматический выбор, композицию и взаимодействие Web-сервисов для выполнения определенной задачи, обусловленной высокоуровневым описанием задания. Например, пользователь хочет спланировать все мероприятия, связанные с поездкой на конференцию. Обычно пользователь должен выбрать все нужные Web-сервисы, вручную задать порядок их выполнения и удостовериться, что все программное обеспечение, необходимое для взаимодействия, соответствует его требованиям. Но если Web-сервис размечен при помощи OWL-S, то информация, необходимая для выбора и композиции сервисов, представлена на их сайтах. Соответствующее программное обеспечение может манипулировать этими данными, а также спецификациями целей задачи, для автоматического выполнения задачи. Таким образом, OWL-S должен обеспечивать декларативные спецификации предварительных условий и результатов выполнения отдельных сервисов, необходимые для автоматической композиции и взаимодействия сервисов.

2. **Автоматический мониторинг выполнения Web-сервисов**. Отдельные сервисы, и, более того, композиции сервисов, обычно требуют некоторого времени для выполнение. Пользователь может захотеть в этот период узнать, каков статус его запроса, или же его планы могут измениться. Например, пользователь хочет удостовериться в том, что резервирование отеля успешно выполнено. Для этого было бы полезно иметь возможность выяснять, на каком этапе находится процесс выполнения запроса и не возникли ли какие-либо непредвиденные помехи. Таким образом, OWL-S должен обеспечивать декларативные описания для состояния выполняемых сервисов.

Онтология ПрО и Web-сервисы

Наличие явного представления знаний о ПрО, к которой относится сервис, допускает переформулировку запросов контекстно-зависимым способом и переговоры о возможностях этого сервиса. Наиболее распространенным на сегодня механизмом представления знаний о ПрО являются онтологии [5]. Одна и та же ПрО может иметь несколько онтологий, поскольку информация о ПрО доступна даже экспертам

лишь частично. Использование при поиске семантики ПрО обеспечивает более корректную компоновку Web-сервисов. Описывая определенный Web-сервис для публичного использования, его провайдеры должны связывать его с определенными онтологиями, которые позволяют формализовать терминологию соответствующей ПрО. Например, назначение сервиса описано как "оценка квартиры". При этом во входных параметрах используются такие термины, как адрес квартиры, метраж и количество комнат. Но пользователю необходимо пояснить, что "метраж" - это "общая площадь", а не "жилая площадь". Другой пример - при обращении к сервису покупки автомобиля или компьютера пользователю надо предложить структуру параметров (таксономию либо более сложную конструкцию), характеризующих это устройство (причем приводить несколько видов альтернативных названий). При этом можно соотносить друг с другом сервисы различных провайдеров для планирования последовательности их совместного использования.

Алгоритмы нахождения соответствия между запросом и сервисом, использующие онтологическое представление знаний, позволяют автоматизировать нахождение семантического подобия между запросом и описанием сервиса, несмотря на синтаксические различия между ними. Для этого запрос согласовывается на основе иерархии понятий ПрО, отображенной в онтологии [6]. Например, запрос об автомобилях соответствует объявлению о транспортных средствах, так как автомобили включены в категорию "транспортные средства". Соответствие между описанием Web-сервиса и запросом обнаруживается, когда все выходы запроса согласованы с выходами описания, и все входы описания – со всеми входами запроса, то есть когда сервис способен удовлетворить потребности запрашивающей стороны, и запрашивающая сторона обеспечивает все входы согласованных сервисных потребностей в его действиях.

Формирование онтологий ПрО

Изложенные выше подходы оставляют открытыми вопросы о том, кто и каким образом формирует онтологии ПрО, с понятиями которых связаны имена параметров Web-сервиса, и как строится описание Web-сервиса в OWL-S (разработчики и провайдеры Web-сервисов не обязаны владеть онтологическим анализом и знать инструментальные средства создания онтологий). В связи с этим актуальной задачей представляется разработка средств и методов автоматизированного формирования онтологий по информационным ресурсам, соответствующим определенному Web-сервису.

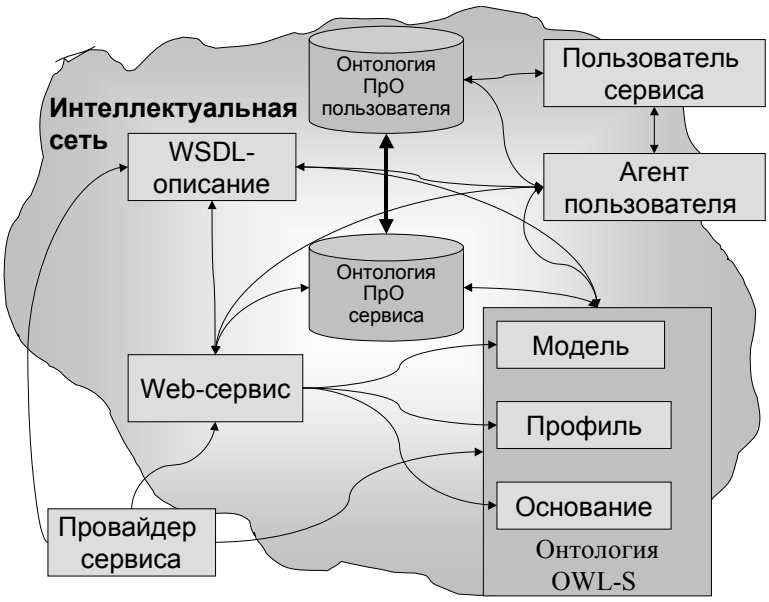


Рис.1. Средства задания семантики Web-сервиса

Важным вопросом является также создание общего словаря (тезауруса) ПрО, обеспечивающего взаимопонимание пользователей и разработчиков Web-сервисов. Кроме того, очень актуальна разработка

эффективных алгоритмов сравнения онтологий, которые, возможно, являются различными концептуализациями одной и той же ПрО – для нахождения соответствия между онтологиями пользователей и разработчиков Web-сервисов. В ряде случаев пользователи и разработчики могут воспользоваться готовыми онтологиями, но поиск таких онтологий также является нетривиальной задачей при нестандартном терминологическом базисе. Иногда ПрО приложения настолько специфична, что требует значительного уточнения и расширения для уровня приложения самим пользователем. Предлагается использование методов индуктивного обобщения для автоматизированного извлечения онтологических знаний о ПрО из набора информационных ресурсов. релевантных этой ПрО.

Методы сопоставления онтологий

При наличии онтологических описаний как Web-сервисов, так и запросов пользователей возникает проблема сравнения этих описаний. Если Web-сервисы и запросы ссылаются на одну онтологию, то можно легко установить, связаны имена параметров атомарных сервисов с одним понятием онтологии или с разными. В противном случае необходимо установить, являются ли понятия различных онтологий эквивалентными (например, синонимами) или находятся в иерархических отношениях (например, являются подклассом). В общем случае эта довольно сложная задача, которая имеет высокую вычислительную сложность. Наш подход к ее решению базируется на предположении, которое в поиска информации используются относительно небольшие и простые за структурой фрагменты онтологий, которые характеризуют семантику конкретных ИР и запросов пользователей. Алгоритм сравнения таких онтологий состоит из следующих этапов:

1. Строится пересечения терминов онтологий Web-сервиса и запроса $T(O) = T(O_s) \cap T(O_q)$.

2. Если это пересечение не пусто, для каждого термина из T(O) строятся два множества T_s и T_q- термины, которые связанные с ним в каждой онтологии любыми отношениями.

3. Для каждого термина из T(O) строится пересечение множеств T_s и T_q.

4. Анализ типов отношений между терминами из T(O) и пересечения множеств T_s и T_q (все отношения онтологии делятся на три типы - иерархические, синонимические и прочие).

5. Строится коэффициент сходства онтологий, который является количественным отображением сходства семантики двух онтологий. При этом учитываются следующие факторы: вхождение одного и того же термина в обе онтологии; то, что два термина находятся в разных онтологиях в одном и том же отношении; то, что два термина находятся в разных онтологиях в отношениях одного типа или разных (например, в иерархическом отношении и отношении синонимии); существуют ли вообще любые отношения (прямые или опосредствованные) между одними и теми же терминами. Для этого используются статистические методы, нечеткую логику, интенсиональные отношения и эмпирические правил.

6. Строится коэффициент подобия запроса и Web-сервиса – аналогично п.5, но учитываются только термины из $T(O) = T(O_s) \cap T(O_q)$, на которые ссылаются имена параметров Web-сервиса. Если полученный коэффициент выше определенного пользователем коэффициента доверия, то считается, что Web-сервис удовлетворяет потребностям пользователя и может использоваться при компоновке составного Web-сервиса.

Выводы

Рассмотрев базовые составляющие сервис-ориентированных вычислений в распределенной среде Интернет и проанализировав перспективы их развития, можно сделать выводы о том, что автоматизация

компоновки Web-сервисов, которая должна обеспечить их значительно более широкое применение, должна базироваться на семантическом описании их функциональных возможностей. Сегодня описание семантики Web-сервисов, как и многих других информационных ресурсов распределенной гетерогенной среды Интернет, связывают с онтологическим подходом к представлению знаний (OWL-S). Однако открытыми остаются вопросы как создания онтологий, адекватно отражающих специфику определенных ПрО, так и проблемы, связанные со сравнением и установлением соответствий между различными онтологиями. В данной работе предложены подходы к установлению подобия между онтологиями, характеризующими Web-сервисы и потребности пользователей .

Литература

1. Cowles P. Web Services and the Semantic Web. – http://www.sys-con.com/webservices/article.cfm?id=419.
2. Christensen E., Curbera F., Meredith G. Web services description language (WSDL) 1.1, 2001 www.w3.0rg/TR/wsdl.
3. Гладун А.Я., Несен М.В., Штонда В.Н. Интеллектуальные агентно-ориентированные услуги, базирующиеся на платформах интеллектуальных сетей// Компьютерные средства, сети и системы, 2004, №6, с. 112-122.
4. OWL-S: Semantic Markup for Web Services. – http://www.daml.org/services/owl-s/1.0/owl-s.html.
5. Клещев А.С., Артемьева И.Л. Отношения между онтологиями предметных областей. Ч. 1. // Информационный анализ, Выпуск 1, С.2, 2002. – С.4-9.
6. Рогушина Ю.В., Гладун А.Я. Онтологический поход к мультилингвистическому анализу информационных ресурсов в сети Интернет // Сб.трудов VI международн.конф. "Интеллектуальный анализ информации ИАИ-2006", К.: Просвіта, 2006. – С.237-246.

Информация об авторах

Гладун Анатолий Ясонович – *Международный научно-учебный центр информационных технологий и систем НАНУ, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40, email: glanat@yahoo.com*

Рогушина Юлия Витальевна – *Институт программных систем НАНУ, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40, Киевский Славистический Университет, email: _jij_@ukr.net*

Штонда Виктор Николаевич – *Издательство компьютерной литературы «Диалектика», Киев-157, 03412, просп. акад. Глушкова, 4, Киевский Государственный Университет им. Т.Г.Шевченко, email: shtonda@dialektika.com*

ИНСТРУМЕНТАЛЬНАЯ СРЕДА XG#: ОНТОЛОГИЧЕСКИЙ ПОДХОД К РАЗРАБОТКЕ ИНТЕГРИРОВАННЫХ ОБОЛОЧЕК ЭКСПЕРТНЫХ СИСТЕМ

Михаил Никулин

Аннотация*: В статье описывается подход к реализации инструментальной среды XG# для создания оболочек экспортных систем с интегрированным представлением знаний. Основной особенностью среды XG# является использование онтологического подхода как для представления прикладных знаний уровня пользователя-эксперта, так и базовых примитивов, выступающих в качестве каркаса ядра инструментальной среды. В частности, онтологии используются для представления входного языка системы, что позволяет не только адаптировать входной язык под пользователя-эксперта, но и интегрировать различные средства представления знаний посредством внесения дополнений и*

изменений в исходную грамматику. Предлагаемый подход позволяет вносить изменения в грамматику входного языка без переписывания уже существующего кода системы, а лишь путем добавления и регистрации новых библиотек, реализующих требуемую функциональность.

***Keywords**: экспертные системы, метазнания, онтологический подход, проектирование и разработка информационных систем.*

***ACM Classification Keywords**: I.2 Artificial Intelligence: I.2.4 Knowledge Representation Formalisms and Methods – Representation languages; I.2.5 Programming Languages and Software – Expert system tools and techniques*

Введение

Современные оболочки экспертных систем (ЭС), особенно универсальные оболочки ЭС, являются весьма дорогостоящим продуктом, а среди оболочек, наиболее подходящих для целей обучения и, в тоже время, являющихся практически значимыми при разработке реальных экспертных систем, отсутствуют средства, позволяющие адекватно учитывать специфику проблемной области и стиль рассуждений конкретного эксперта, его предпочтений с точки зрения требований к языку представления знаний (ЯПЗ). Поэтому представляется весьма актуальной задача [Chuprina, 2005] разработки современной объектно-ориентированной инструментальной среды создания оболочек ЭС с учетом требований, предъявляемых пользователями к средствам представления знаний.

Данная статья посвящена описанию концепции построения и реализации системы XG#, которая направлена на решение указанной выше проблемы. В основе разрабатываемой инструментальной среды лежит специальный *интерфейс разработчика*, позволяющий в наглядной форме с использованием визуальных средств так называемого *инспектора грамматик* [Nikulin, 2004] вносить изменения в описание синтаксиса языка представления знаний, его семантики, а также определять прагматический аспект языковой конструкции относительно его интерпретации в терминах базовых примитивов системы. Онтология, описывающая входной язык системы, является изменяемой частью онтологической базы метазнаний, называемой *метасхемой* XG#, описывающей поведение системы. Компоненты среды XG# реализуются в терминах базовых примитивов ядра и способны автоматически настраиваться на содержимое метасхемы. Предлагаемый подход делает возможным эволюционное развитие системы и позволяет наращивать ее возможности путем подключения новых компонент, предназначенных, например, для интеграции различных средств представления знаний (продукций, фреймов и др.).

Архитектура XG#

Обобщенная архитектура инструментальной среды XG# показана на рис. 1. Центральной компонентой является метасхема, которая содержит:

- *онтологию базовых примитивов* (знания системы о самой себе);
- *онтологию входного языка системы* (синтаксис, семантика, прагматика);
- *тиражируемые проблемно-ориентированные онтологии понятий.*

Часть метазнаний системы, а именно, *системные метазнания*, автоматически формируется самим ядром виртуальной машины XG#. Для создания и пополнения той части базы метазнаний, которая описывает язык представления знаний разрабатываемой оболочки ЭС, создан специальный визуальный интерфейс разработчика, так называемый "Инспектор грамматик". *Инспектор грамматик* позволяет в наглядной форме описывать и динамически изменять лексику, синтаксис, семантику и правила генерации объектного кода языка представления знаний.

Таким образом, инструментальная среда XG# – это своего рода *универсальная оболочка экспертных систем, проблемной областью которой является* создание специализированных *оболочек ЭС*, учитывающих специфику проблемной области, с определенным, возможно, интегрированным средством представления знаний и механизмом функционирования. Универсальность достигается не за счет того, что в систему изначально зашиты в интегрированном виде все основные средства представления знаний с поддержкой соответствующих механизмов логического вывода, а за счет возможности динамического внесения изменений в грамматику языка представления знаний.

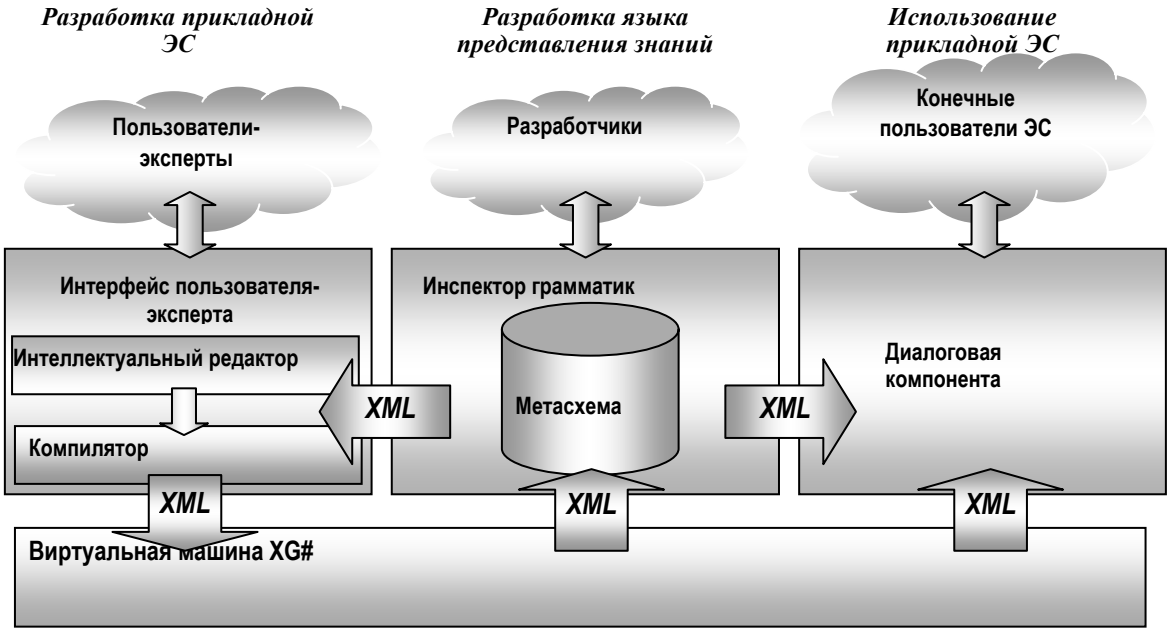


Рис.1. Архитектура XG#

Компилятор системы XG# получает необходимую информацию о структуре языка из метасхемы. Основной принцип работы компилятора – *раздельное проведение основных фаз трансляции исходного текста программы.* Фаза *синтаксического анализа* формирует, согласно метасхеме, дерево синтаксического разбора программы либо выдает сообщения об обнаруженных ошибках. На фазе *семантического разбора* происходит анализ семантической корректности дерева синтаксического разбора, согласно правилам, определенным в метасхеме. *Генератор кода* на входе получает семантически правильное дерево синтаксического разбора программы в виде XML и преобразует его с помощью XSL к XML объектного кода виртуальной машины.

Интеллектуальный редактор XG# предназначен для редактирования правил объектного уровня и метаправил уровня предметной области. Интеллектуальным редактор назван потому, что он способен настраиваться на изменения во входном языке системы и поддерживает современный уровень IntelliSense (подсветка синтаксиса, выдача подсказок пользователю). Это стало возможным благодаря тому, что информацию о грамматике конкретного входного языка он берет непосредственно из метасхемы. Кроме того, для генерации контекстных подсказок в процессе редактирования исходного текста программы интеллектуальный редактор использует компилятор для неполного грамматического разбора фрагментов текста. Это дает возможность, например, вывести адекватный текущему контексту редактирования список допустимых конструкций или набор допустимых значений из домена соответствующей переменной в посылке продукции.

Основное назначение виртуальной машины – выполнение объектного кода. Ядро виртуальной машины способно выполнять базовые арифметические операции, операции по приведению типов и т.п. Все дополнительные компоненты (механизмы логического вывода, библиотеки функций) реализованы в виде отдельных модулей, которые подключаются к ядру системы через специальный шлюз. Для библиотек с компонентами, в свою очередь, также определен интерфейс, посредством которого ядро "опознаёт" и подключает соответствующие модули.

Базовые примитивы XG#

В настоящее время понятие «онтология» является одним из наиболее часто используемых понятий. Термин «онтология» применяется в различных контекстах, в которых ему приписывается различный смысл. Учитывая специфику решаемых в данной работе задач, будем считать, что онтология – это точная спецификация некоторой области, которая включает в себя словарь терминов (понятий) предметной области и множество связей между ними (типа «элемент-класс», «часть-целое»), которые описывают, как эти термины соотносятся между собой в конкретной предметной области. Фактически в данном случае *онтология – это иерархическая понятийная основа рассматриваемой предметной области*, для которой разработана информационная система.

Онтология базовых примитивов ядра XG# основывается на простейших понятиях «сущность» и «связь». Все объекты системы представляются *множеством взаимосвязанных сущностей*. Объект «связь» может выступать как в роли атрибута, связывая при этом сущность и значение, так и в роли отношения, связывая две или более сущности. Каждая сущность описывается онтологией, состоящей из других сущностей. Таким образом, сущности выступают в роли «классов» или и в роли «экземпляров». Роль «класса» для сущности типа класс играют особые сущности «классы», называемые «мета-классы». Мета-классы являются самоописывающимися сущностями, выступающими в качестве экземпляров и классов для самих себя одновременно. Такое замыкание онтологии символизирует количество осмысленных метауровней и исключает необходимость бесконечного описания каждой мета-сущности на новом мета-мета-уровне. Отношения сущностей и их атрибуты описываются специальными сущностями-экземплярами называемыми «связь». Сущность «связь» описывает арность и классы связываемых сущностей посредством других связей с соответствующими классами сущностей. Поскольку каждая связь описывается соответствующей ей сущностью «связь», в конечном итоге найдется сущность «связь», которая будет описывать такие типы связей, которыми она сама и представлена в онтологии базовых понятий XG#. Замкнутость и самоописываемость понятий являются такими характерными особенностями онтологии ядра XG#, которые, по сути, отражают знания разработчиков об устройстве системе XG#. Это позволяет развивать и эволюционно наращивать функциональность компонентов системы.

Кроме того, возможность динамического изменения грамматики входного языка позволяет безболезненно расширять функциональность системы без переписывания уже существующего программного кода ядра системы. Так, например, для адаптации языка под расширенные возможности продукционного механизма логического вывода в части представления метаправил, необходимо было дать возможность пользователю корректно описывать метаправила на языке XG# и в дальнейшем осуществлять генерацию кода метаправил понятного для виртуальной машины.

До модификации компоненты вывода грамматика языка была рассчитана лишь на представление правил объектного уровня и выглядела следующим образом (рис. 2.):

<программа>::= [<описание доменов>] <описание переменных> <эс>;

<эс>::= <описание цели> [<блок инициализации>] [<блок заключения>] <описание правил>;

<описание правил>::= rules <список правил>;

<список правил>::= <правило>;

<список правил>::= <правило> <список правил>;

<правило>::= rule : Идентификатор <посылка> <заключение>;

Для описания метаправил грамматика языка системы XG# была расширена (см. рис. 3):

<программа>::= [<описание доменов>] <описание переменных> <эс>

<эс>::= … <описание правил> <описание мета-правил>

<описание правил>::= rules <список правил>

<список правил>::= <правило>

<список правил>::= <правило> <список правил>

<правило>::= rule : Идентификатор <посылка> <заключение> [<класс правила>]

<описание метаправил>::= metarules <список метаправил>

<список метаправил>::= <метаправило>

<список метаправил>::= <метаправило> <список метаправил>

<метаправило>::= metarule : Идентификатор <посылка> <заключение>

Рис. 2. Грамматика продукционного правила

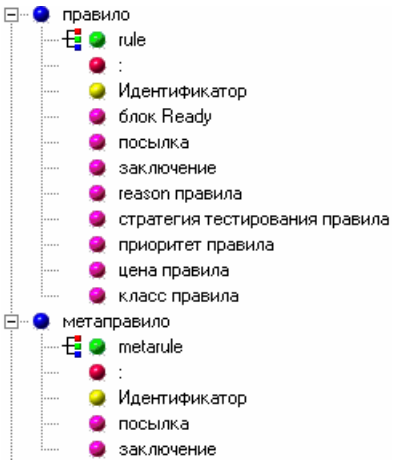


Рис. 3. Грамматика продукционного мета-правила

Таким образом, модификация онтологии входного языка XG# позволила без переписывания уже существующего кода компонент системы расширить функциональность механизма логического вывода. Учитывая ориентацию системы на цели обучения, наличие инспектора грамматик (визуального инструмента для явного представления и редактирования онтологии входного языка), служит целям более глубокого осмысления студентами способов использования метазнаний.

Заключение

В данной статье был рассмотрен подход к реализации инструментальной среды создания оболочек экспертных систем XG#, в которую была заложена возможность динамического изменения грамматики входного языка системы, с целью обеспечения адаптируемости системы под потребности пользователя-эксперта, проблемную область, для целей обучения, а также с целью её дальнейшей расширяемости. Таким образом, описанная в данной работе архитектура настраиваемого компилятора позволила разработать среду-оболочку для создания оболочек, входным языком которых могут быть не только языки представления знаний в общем случае, но и широкий спектр других строго типизированных языков, грамматика которых удовлетворяет требованиям, описанным работе [Nikulin, 2004].

Библиографический список

[Nikulin, 2004] М.Б. Никулин, А.В. Жигалов. Концепция построения и реализации оболочек экспертных систем на базе метазнаний. В сб. тр. V Всерос. научно-практ. конф. молодых ученых, аспирантов, студентов "Молодежь. Образование. Экономика". Ч.4. Ярославль: Изд-во "Ремдер", 2004. С. 81-86.

[Chuprina, 2005] С.И. Чуприна, М.Б. Никулин. Инструментальная среда XG#: разработка проблемно-ориентированных оболочек экспертных систем с интегрированным представлением знаний // Успехи современного естествознания. №5, 2005. Материалы XXXII Междунар. конф. IT+S&E'2005 "Информационные технологии в науке, образовании, телекоммуникации и бизнесе". Украина, Гурзуф, 2005. С. 105-107.

Сведения об авторе

Михаил Никулин *– Пермский государственный университет, аспирант кафедры математического обеспечения вычислительных систем; Россия, г. Пермь, 614990, ул. Букирева, д. 15; e-mail: mihmix@mail.ru*

КОНЦЕПТУАЛЬНЫЕ ПРИНЦИПЫ РЕАЛИЗАЦИИ И СТРУКТУРА ИНСТРУМЕНТАРИЯ КОНТРОЛЯ ЗНАНИЙ НА БАЗЕ ОНТОЛОГИЙ

Елена Нетавская

Аннотация*: В статье выполнен анализ современных инструментальных средств для проведения контроля знаний. Предложена концепция создания и принципы реализации инструментария контроля знаний. Указано на особенности формирования элементного базиса и предпосылки формирования базы знаний. Разработаны алгоритмы работы с экспертной системой контроля знаний как для преподавателя, так и для экзаменуемого. Определены алгоритмические особенности проведения контроля знаний с использованием логической схемы курса и онтологии предметной области.*

Ключевые слова*: Контроль знаний, учебный процесс, онтологии, логическая схема курса.*

Введение

Предметом нашего исследования является процесс обучения и контроля знаний студентов с помощью экспертных систем. Очевидно, что в подавляющем большинстве случаев контроль знаний сильно субъективизирован как по форме его проведения, так и по содержанию. Его объективизация может быть достигнута и достигается с помощью использования автоматизированных систем. Однако при таком подходе не гарантируется полнота охвата учебного материала, качество его представления для контроля, кроме того время проведения контроля не является оптимизированным, а его процесс сопровождается информационной избыточностью.

Современные системы автоматизированного контроля знаний, по нашему мнению, можно разделить на две категории, реализованные:

− без использования онтологий;

− с использованием онтологий.

В свою очередь, в основном системы из первой категории базируются на таких основных идеях и подходах к оцениванию знаний:

− вопросы имеют вид тестов, с двумя или больше вариантами ответов; вопросы задаются в определенной или случайной последовательности; оценка определяется как отношение количества правильных ответов к количеству всех вопросов;

− вопросы имеют вид тестов; вопросы задаются в случайном порядке, но случайность определяется вероятностями актуальности того или иного вопроса (под актуальностью понимают наличие или отсутствие связи между настоящим и предыдущим вопросами, сложность вопроса и т.п.); общая оценка определяется как функция от "взвешенных" вопросов;

− вопросы классифицированы по типам; они задаются случайным образом, но обязательно указанное количество вопросов определенного типа; для каждого типа вопросов существуют процедуры оценки и общая оценка является интегральным показателем.

Онтологии в учебном процессе используются пока редко, что связано с большой трудоемкостью процесса их формирования и процедурой использования для тестирования. Известны такие подходы:

− обучаемый составляет из вопросов целостную картину предмета с указанием концептов и отношений между ними;

− на каждом шаге обучаемому предлагается несколько вопросов, из которых он выбирает один, наиболее на его взгляд частный, и на него отвечает, после чего переходит к более общему.

В работе [Gruber, 1995] онтология определена как точная спецификация концептуализации. Разные точки зрения на такое определение обсуждаются в [Guarino, 1998]. В частности, обозначены онтологии верхнего уровня, онтологии предметной области, онтологии задач и онтологии приложений. Значительное количество работ посвящено проблеме использования онтологий в электронном обучении, их обзор приведен в [Henze, 1997]. Анализ релевантных публикаций свидетельствует о том, что идея применения онтологий в учебном процессе сосредоточена на повышении качества обучения. Технологии контроля знаний с использованием онтологии предметной области находятся в инициальной стадии разработки.

Один из методов контроля знаний на базе онтологии заключается в том, что оценивание результатов такого тестирования происходит в зависимости от того, насколько достигнута цель контроля, т.е. правильно ли построена логическая цепочка вопросов и отвечает ли она онтологии. Рассмотренные варианты построения экспертных систем не обладают свойствами полноты представления и контроля учебного материала, имеют информационную избыточность и не оптимизированы по критерию минимизации времени проведения контроля. Большинство перечисленных недостатков отсутствуют в экспертных системах, базирующихся на использовании композиции логической схемы курса и онтологии предметной области, предложенных в работах [Netavskaya, 2007]. В настоящей статье рассмотрим конструктивные особенности построения инструментальных средств оценивания знаний.

Концепция создания и принципы реализации инструментария контроля знаний

При разработке инструментария контроля знаний на базе онтологии предметной области рационально использовать два подхода: проблемно-ориентированный (целеориентированный) и ориентированный на доминирующее изучение моделей и методов (объектно-ориентированный). В основе каждого из них лежит схема изложения материала учебного курса. Для первого подхода характерна такая последовательность:

$$P \to < Z_1, Z_2, ..., Z_n > \to < M_1, M_2, ..., M_k > \to < T_1, T_2, ..., T_l > \to < S_1, S_2, ..., S_m >, \quad (1)$$

где P – проблема, $Z_1, Z_2, ..., Z_n$ – соответствующие ей задачи, $M_1, M_2, ..., M_k$ – модели, $T_1, T_2, ..., T$ – методы, $S_1, S_2, ..., S_m$ – средства решения указанных задач. Такой подход имеет нисходящий характер. В другом случае логическая цепочка представления учебного материала будет такой:

$$< M_1, M_2, ..., M_k > \rightarrow < T_1, T_2, ..., T_l > \rightarrow < Z_1, Z_2, ..., Z_n > \rightarrow < P_1, P_2, ..., P_q > \rightarrow < S_1, S_2, ..., S_m > . \quad (2)$$

В этом случае в начале курса излагается определенная совокупность моделей и методов, далее рассматриваются задачи, которые решаются с использованием уже изученных моделей и методов и которые сами являются составляющими частями множества проблем. Заканчивается учебный курс рассмотрением соответствующих инструментальных средств. В дальнейшем изложении будем базироваться на представлении (1).

Предлагаемая концепция построения экспертных систем для контроля знаний содержит теоретические и практические составляющие. В частности:

– идея и необходимость построения экспертных систем контроля знаний определяется задачей повышения эффективности процессов обучения и контроля знаний;

– объективизация процесса контроля достигается посредством использования экспертных систем;

– достаточно точная оценка гарантируется процедурой, обеспечивающей полноту представления материала в процессе контроля знаний, а также его отображением на структуру множества вопросов;

– информационная избыточность устраняется с помощью алгоритма, определяющего в режиме реального времени необходимость и структуру дальнейшего контроля знаний;

– все вышеперечисленные факторы направлены на минимизацию времени оценивания.

Элементный базис и структура экспертной системы определяется необходимостью:

– работы с ней как лица, проходящего контроль знаний, так и эксперта (системного аналитика, преподавателя, лица, принимающего решение);

– создания базы знаний, содержащей концептуальные элементы курса и отношения между ними, представленные в виде графа “И-ИЛИ”;

– разработки онтологии предметной области, содержащей ее концепты, отношения между ними и их интерпретации;

– формирования базы данных, содержащей вопросы и варианты ответов для проведения контроля знаний;

– разработки процедуры формирования последовательности вопросов и оценивания разных типов вопросов, а также интегральной оценки.

Реализация инструментальных средств должна базироваться на таких принципах:

1. Принцип ясности. Все концепты, факты, отношения, интерпретации, представленные в структурных элементах экспертной системы должны иметь однозначное трактовку на естественном языке, несмотря на формализмы, в них присутствующие [Gruber, 1993].
2. Принцип универсальности. В системе должна быть предусмотрена возможность формирования онтологии и логической схемы задач по разным курсам.
3. Принцип согласованности. Все концепты, полученные в результате точного логического вывода из аксиом не должны противоречить неформальным определениям и примерам.
4. Принцип расширяемости. Необходимость введения новых концептов не должна подвергать ревизии структуру уже существующих дефиниций.

5. Принцип минимальности смещения кодирования. Выбор представления не должен иметь влияния на качество определений или следствий.
6. Принцип открытости: предусмотрена возможность внесения изменений и дополнений как в модули системы, так и в элементный базис и структуру онтологий и логической схемы задач.

Инструментарий для проведения контроля знаний должен предусматривать работу двух категорий пользователей: экзаменатора и экзаменуемого. Экзаменатор должен уметь:

− формировать онтологии, а именно определять концепты (в композиции автоматического режима с ручным или исключительно ручном), отношения, составлять словарь интерпретаций и его представление;

− формировать логическую схему курса;

− формировать базу вопросов и ответов с комментариями (помощью, подсказками для экзаменуемого);

− определять процедуру оценивания ответов (необходимо предусмотреть определения коэффициентов важности вопросов с разработкой процедуры их модификации в процессе тестирования, например – чем больше экзаменующихся не ответили на вопрос – тем выше его важность).

Для экзаменуемого достаточно знать процедуру работы с экспертной системой.

Элементный базис и предпосылки формирования базы знаний

В качестве иллюстрации рассмотрим формирование базы знаний для учебного курса “Основы автоматизированного проектирования сложных объектов и систем”. Главной проблемой, которой посвящено изложение материала, является повышение эффективности процесса проектирования путем использования человеко-машинных систем. Для ее решения используется несколько известных концепций: модульное проектирование (МП), в котором главное внимание уделено эффективному выполнению отдельных задач; объектно-ориентированное проектирование (ООП) [Booch, 1994], во главу которого поставлены объекты проектирования и отношения между ними; системное проектирование, акцентированное на оптимизации самого процесса проектирования [Тимченко, 1991]. В основе последней концепции лежит такое определение.

Системное проектирование – процесс получения проекта системы в базисе системных свойств, системных ресурсов и структур жизненного цикла.

На рис. 1 представлено фрагмент логической схемы курса, имеющего проблемно-ориентированное построение. Соответствующая структура имеет вид графа «И-ИЛИ». Особенностью такого графа является наличие большого количества элементов с отношением дизъюнкции на нижних уровнях и конъюнкции − на верхних. Логическая схема курса соответствует последовательности изложения материала преподавателем. Ее графовая модель имеет иерархическую структуру, что дает основание для проведения контроля знаний по уровням и этапам (вширь и в глубину). Уровневый контроль позволяет определить глубину знаний экзаменуемого по отдельному учебному элементу, каким может быть тема, задача, модель, метод, приложение, некоторый атрибут и т.п. Этапный контроль предусматривает анализ знаний различных элементов обучения, имеющих одинаковую семантическую нагруженность (например, методы оптимизации – дискретной и непрерывной, приложения для анализа данных − Matlab и Mathcad). Контроль в глубину необходим для уточнения оценки знаний, в то время как контроль вширь используется для предварительного оценивания и определения необходимости дальнейшего оценивания.

Стратегию проведения контроля знаний определяет преподаватель, однако в большинстве случаев рационально придерживаться такой последовательности операций:

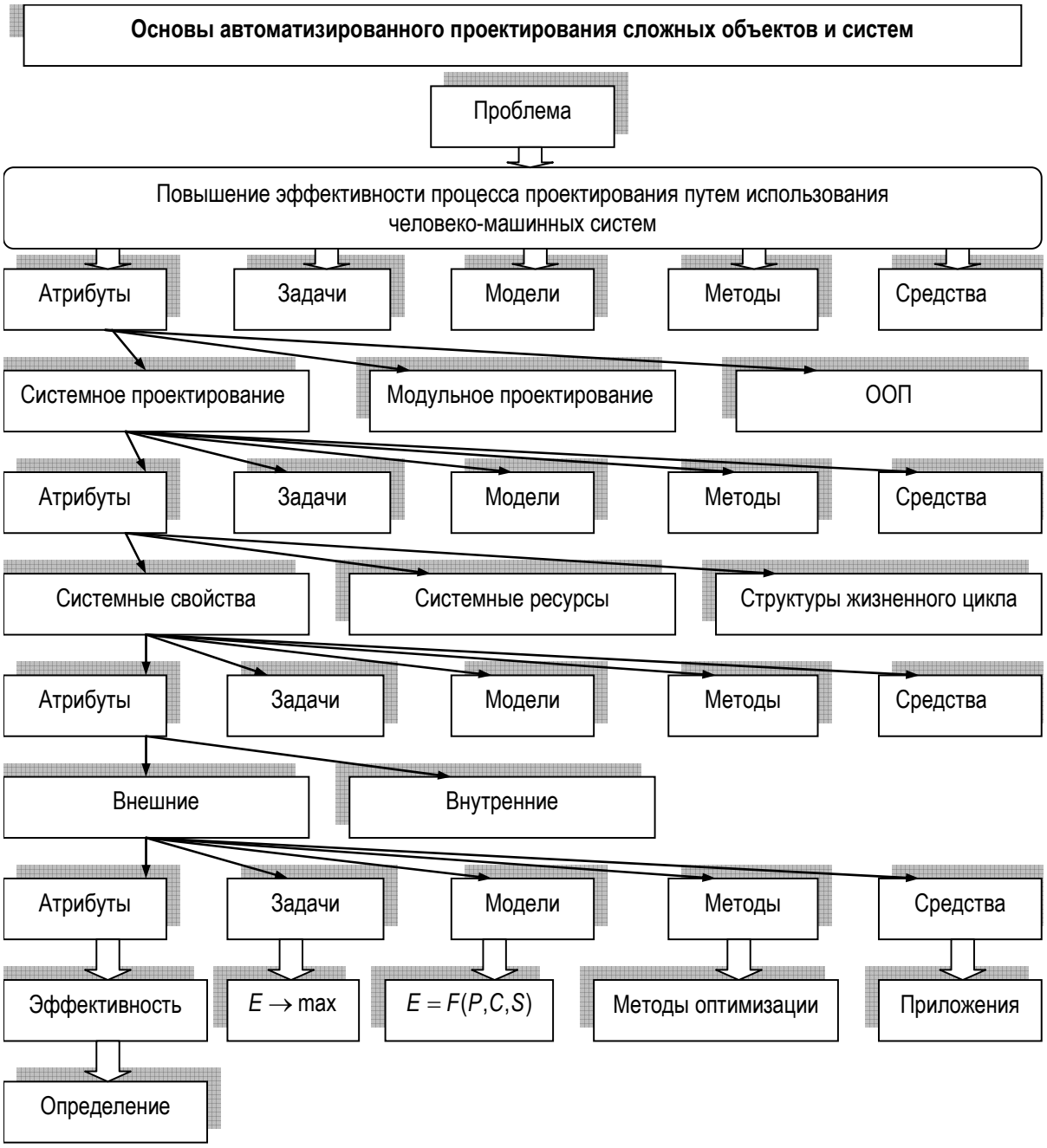


Рис.1. Фрагмент логической схемы задач курса

Шаг 1. Первый вопрос − выбирается случайным образом.

Шаг 2. Анализ ответа. Если ответа нет, его оценка или суммарная оценка является недостаточной для продолжения контроля знаний, то переход на шаг 6. Если оценка достаточно высока, то переход на вопрос обязательно другого, случайно выбранного этапа, если нет − переход на вопрос этого же этапа, но низшего уровня.

Шаг 3. Если выполнено условие остановки алгоритма, то переход на шаг 7.

Шаг 4. Если достигнут последний вопрос в логической схеме контроля, то случайный переход на любой вопрос, но другой ветки.

Шаг 5. Переход на шаг 2.

Шаг 6. По решению преподавателя (формализованному алгоритмически) оценка является неудовлетворительной и переход на шаг 7, либо еще один раз выбрать случайным образом вопрос и перейти на шаг 2.

Шаг 7. Конец алгоритма. Вывод результатов.

Рис. 2. Фрагмент онтологии

То, как формулировать вопрос определяется, исходя из структуры онтологии, соответствующий фрагмент которой изображен на рис. 2. Так, например, установим соответствие между системным атрибутом «Эффективность», приведенным в логической схеме курса и соответствующим концептом онтологии. Если на одном из шагов вышеприведенного алгоритма необходимо задать вопрос, связанный с эффективностью, то он может быть одним из таких:

1. Является ли истинным утверждение: «Численным выражением эффективности является значение критерия эффективности»? Возможные ответы: «Да, Нет». В этом случае критерий эффективности является *атрибутом* понятия эффективности.
2. Какие из перечисленных показателей включает в себя понятие эффективности? Возможные ответы: «рентабельность, фондовооруженность, управляемость, энергоемкость, наблюдаемость, себестоимость». В этом случае потенциальные ответы находятся в отношении «*включает в себя*» с понятием эффективности.

3. Определение понятия эффективность включает в себя такие тезисы: «эффективность − среднее значение качества системы; эффективность − отображение функции построения системы; эффективность − определяющая характеристика процесса проектирования». Приведены ответы, которые находятся в отношении «*часть*» с понятием эффективности.

Заключение

Для процесса контроля знаний характерна значительная субъективность, объективизировать которую стремятся многие преподаватели и экзаменуемые. Необходимо отметить, что такие процессы являются разноаспектными, направления исследования базируются на различных концептуальных парадигмах. В незначительной степени происходит объективизация процесса оценивания. Вместе с тем, теряется полнота охвата учебного материала и разнообразие, выражаемое в семантической сущности задаваемых вопросов. Контроль знаний в большинстве случаев не оптимизирован по времени проведения и смысловой нагруженности вопросов.

В значительной степени устранить указанные недостатки позволяет разрабатываемая автором концепция использования онтологий в процессе контроля знаний. Идеи и принципы, лежащие в ее основе, указывают на композицию четырех составляющих. Первая из них − логическая схема курса, являющаяся базовым элементом при определении последовательности задаваемых вопросов. Вторая − онтология предметной области, предназначенная для формирования вопросов контроля. Классификация вопросов, предусматривающая формализацию вопросов в зависимости от типа ответов, образует третью составляющую. На последнем этапе используется процедура определения интегральной и промежуточных оценок экзаменуемого.

Интеграция указанных элементов позволяет структуризовать учебный материал; выполнить достаточно полное его представление; прерывать контроль знаний в зависимости от условий, определяемых преподавателем; минимизировать информационную избыточность и время тестирования.

Библиография

[Gruber, 1993] T.R. Gruber. A Translation Approach to Portable Ontology Specifications. − Knowledge Acquisition. − 1993. − Vol. 5(2). − Pp. 199-220.

[Gruber, 1995] T.R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. − International Journal Human-Computer Studies. − 1995. − Vol. 43. − Issue 5-6. − Pp. 907-928

[Guarino, 1998] N. Guarino. Formal Ontology in Information Systems. − Proceedings of FOIS'98. − Trento, Italy. −1998. − Pp. 3-15.

[Henze, 1997] N. Henze, P. Dolog, W.Nejdl. Reasoning and Ontologies for Personalized E-Learning in the Semantic Web. − Educational Technology & Society. − 2004. − Vol. 7. − Issue 4. − Pp. 82-97.

[Netavskaya, 2007] E. Netavskaya. Self-Organization of Functioning Process of Expert System with the Use of Subect Domain Ontology. − Information Technologies & Knowledge. − 2007. − Vol. 1. − 219-225.

[Netavskaya, 2007] E. Netavskaya. Inductive approach to forming of control knowledge scheme on the base of subject domain ontology.− Proceedings of ISTA-2007. − Kharkiv, Ukraine. −2007. − Pp. 43-49.

[Booch, 1994] G. Booch. Object-Oriented Design with Applications. Second Edition. The Benjamin/Cummings Publishing Company, Inc, Menlo Park, CA, 1994.

[Тимченко, 1991] А.А. Тимченко, А.А. Родионов. Основы информатики системного проектирования объектов новой техники. – Киев: Наукова думка, 1991. – 152 с.

Информация об авторе

Елена Нетавская – *Черкасский государственный технологический университет, бул. Шевченко, 460, Черкассы, Украина; e-mail: neelena@list.ru*

INFORMATION SEARCH BASED ON ANALYSIS OF EXPERTS STATEMENTS[1]

Gennadiy Lbov, Nikolai Dolozov, Pavel Maslov

***Abstract**: The paper describes natural language processing. The proposed method, which uses statement coordination principle, is implemented to the described search system. This method allows to compile an ordered list of answers to the inquiry in the form of quotations from the document.*

***Keywords**: natural language processing, coordination of statements, information search.*

***ACM Classification Keywords**: H.3.3 Information Search and Retrieval*

Introduction

The specified approach defines criterion of selection of significant sentences of documents, based on accordance to a certain logic structure reflecting the sense of inquiry, also allow revealing relevant documents in the order of inquiry level accordance to the documents.

Coordination of statements

Basing on outcomes of NLP (see below) the logic form is constructed for each sentence. This form is a model in the language of predicates calculus of two variables united in conjunctions. Each of such predicates is an elementary statement. Let X_i, Y_i, Z_i etc. be each predicate variable.

The set (for each type of a predicate), corresponding sentences of the text is a variety of coordinated statements, and a set corresponding inquiry is beforehand coordinated statement. By quantity of the coordinated predicates the level of its accordance to inquiry is defined, being based that each predicate is a part of model of the sentence.

Let some statement with known characteristics requires to define its accordance to inquiry [1]. The general formal writing of a sentence is done in the form of two-place predicates conjunction. We shall designate T_{ji}^{k} as area of the validity of function and argument variables in the initial sentences inquiry, where i, j, k are the numbers of predicates, statements and the links between argument and function variables, respectively. As variables are nominal the area of true statements is defined by variables satisfying the list of admissible values. As such list the dictionary of synonyms is used.

As predicates two-place and their variables are defined on different areas of the validity for the coordination of statements, it is necessary to consider variables in predicates separately.

For each predicates contained in the statement the areas of validity are defined: T_{pi}^{1} is a truthful area of the first variable in the predicate *i* the inquiry *p*; T_{pi}^{2} is the same for the second variable. Let us designate T_{ji}^{1}, T_{ji}^{2} as truthful areas of variables in predicates of the input text. Respectively, the statement satisfying:

[1] The work was supported by the RFBR under Grant N07-01-00331a.

$$1.\quad \frac{\mu(T^2_{jl} \cap T^2_{ql})}{\mu(T^2_{jl} \cup T^2_{ql})} \geq \beta_{r2} \quad and \quad \frac{\mu(T^1_{jl} \cap T^1_{ql})}{\mu(T^1_{jl} \cup T^1_{ql})} \geq \beta_{r1} \quad \text{- true}$$

$$2.\quad \frac{\mu(T^2_{jl} \cap T^2_{ql})}{\mu(T^2_{jl} \cup T^2_{ql})} \geq \beta_{r2} \quad and \quad \frac{\mu(T^1_{jl} \cap T^1_{ql})}{\mu(T^1_{jl} \cup T^1_{ql})} < \beta_{r1} \quad \text{- not likely}$$

$$3.\quad \frac{\mu(T^2_{jl} \cap T^2_{ql})}{\mu(T^2_{jl} \cup T^2_{ql})} < \beta_{r2} \quad \text{- contradictious}$$

$$k = \frac{(N^i_{so})^2}{N^i_s \cdot N_r}$$

Where μ is a cardinal number of the argument; β_{rq} is a parameter is defined experimentally in [1], N^i_s is the number of all predicates in a sentence, N^i_{so} the number of the coordinated predicates of a sentence, N_r the number of predicates of inquiry.

To define the accordance of sentence to inquiry it is necessary to calculate ratio k.

Natural Language Processing

The specified approach of selection of significant sentences of documents has been realized programmatically in search system Internal Search System3 further ISS3 [2] (In this system some technologies of known system [3] are used), which operation scheme is showed in fig. 1, providing search service of documents on local and sharer network resources.

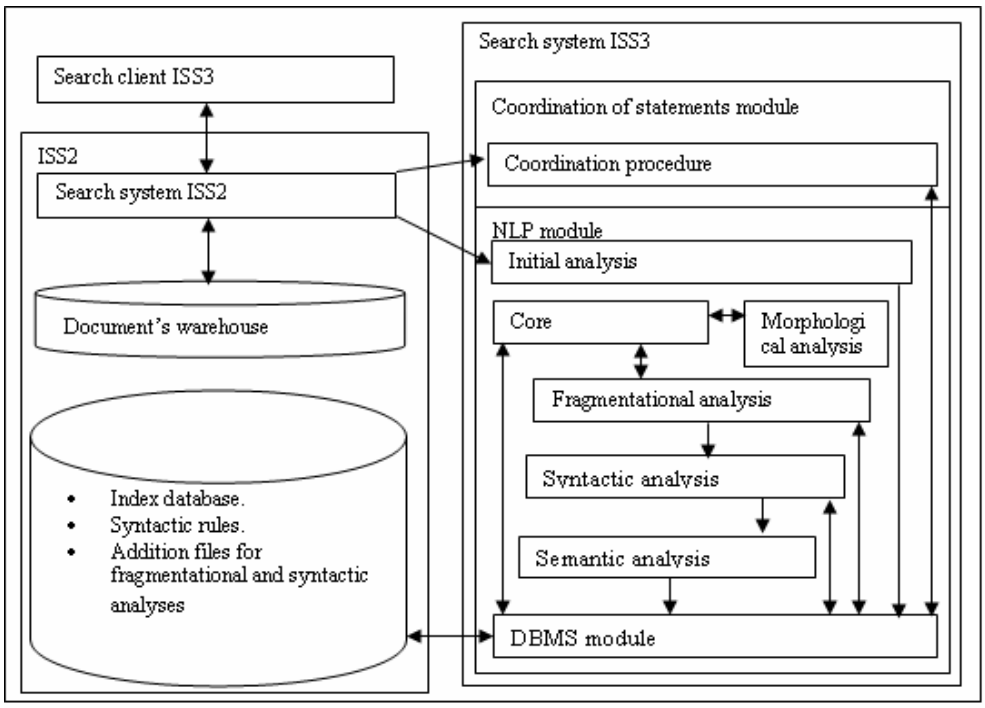


Figure 1. Operation scheme of ISS3.

To perform the subsequent procedure of the coordination of the statement, the developed system represents an input text as a sets of syntactic relations for each sentences of the text. It is achieved by the multilevel natural language processing realizing initial, morphological, fragmentational, and syntactic.

Stages of NL (brief description):

1) Initial text analysis process input text and then forms two separate tables. The first describes elements of the text and their arrangement in the text, and the second defines interrelation of fragments in the input text.

2) At a stage of the morphological analysis, for each lexeme a set of lemmas with attributes is created. Each lemma is represented as a normal form of a word, and attributes - a set of descriptors (a part of speech, number, case etc.).

3) Fragmentational analysis performs derivation of fragment from the input text. Fragments are the main and dependent clauses in structure of complex one, participial, adverbial participial and other isolated turns.

4) The purpose of syntactic parse is automatic construction of a functional tree of a phrase. Syntactic parse process outcomes of morphological and fragmentational analyses.

As a result NLP forms two separate tables for each input document:

1) The content of the document. The table of syntactic relation sets for each sentence of the input document, rows in which describe type and components of syntactic relations.

2) Structure of the document. The table containing the descriptions of the structure of the document (paragraphs, headers, etc.), derived at a stage of the fragmentational analysis and necessary to form the reply to inquiry.

Results

The results are showed below, illustrates processing of various search inquiries in the textual document (fig. 2). For convenience the information of inquiries is contained only in one document, however the system has no limitation for quantity of documents:

```
Тестовый файл.

Человек подошел к столу. Человек взял лист и ручку. Человек начал писать
текст. На столе лежала черная кошка. Кошка заметила человека. Кошка
подбежала к человеку и села на лист.

Рыбак собрался ловить рыбу. Рыбак взял удочку и ведро. Рыбак забросил
крючок в реку и стал ждать. По реке проплывала лодка.
```

Figure 2. Test document.

In accordance with the inquiry "**человек взял ручку**" the title and the first paragraph are extracted:

```
Тестовый файл.

Человек подошел к столу. Человек взял лист и ручку. Человек начал писать
текст. На столе лежала черная кошка. Кошка заметила человека. Кошка
подбежала к человеку и села на лист.
```

The achieved level of conformity of the text to the inquiry, equal to 0.67 is defined by the presence of another syntactic relation "**взять листок**" in the fragment "**Человек взял лист и ручку**".

In accordance with the inquiry "**черная собака**" there is no results returned because the property "**черная**" is related with the element "**кошка**".

These simple tests have shown that the system extracts sentences in accordance with their syntactic structures and the inquiry structure, increasing the search precision in comparison with the offered system without natural language processing.

Conclusion

Approach for performing a search in the textual documents, based on the analysis of syntactic structure of sentences is offered. It allows to extract significant syntactic relation from the text in accordance with the syntactic structure of inquire. As a result of performance of algorithm, sets of the coordinated statements for all types of predicates are formed, each of which describes a certain fragment. To define conformity of the sentence to inquiry, the ratio specified above is calculated.

Bibliography

1. G.S. Lbov, T.I. Luchsheva. The Analysis and Coordination of Expert Knowledge in Problem of Recognition // 2'2004, NAS of Ukraine, pp. 109-112
2. P.P. Maslov. Proceedings of All-Russian scientific conference of young scientists in seven parts. Novosibirsk: NGTU, 2006. Part 1. – 291p. // pp. 250-251
3. The project DIALING // www.aot.ru

Authors' Information

Gennadiy Lbov – *SBRAS, The head of laboratory, full professor, doctor of science; P.O.Box: 630090, Novosibirsk, 4 Acad. Koptyug avenue, Russia; e-mail: lbov@math.nsc.ru*

Nikolay Dolozov *– NSTU, The associate professor, candidate of science; P.O.Box: 630092, Novosibirsk, 20 Marks avenue, Russia; e-mail: dnl@interface.nsk.su*

Pavel Maslov *– NSTU, post-graduate student of Faculty of Applied Mathematic and Computer Science; P.O.Box: 630092, Novosibirsk, 20 Marks avenue, Russia; e-mail: mpp84@rambler.ru*

INTERVAL PREDICTION BASED ON EXPERTS' STATEMENTS*

Gennadiy Lbov, Maxim Gerasimov

Abstract*: In the work [1] we proposed an approach of forming a consensus of experts' statements in pattern recognition. In this paper, we present a method of aggregating sets of individual statements into a collective one for the case of forecasting of quantitative variable.*

Keywords*: interval prediction, distance between expert statements, consensus.*

ACM Classification Keywords*: I.2.6. Artificial Intelligence - knowledge acquisition.*

Introduction

Let Γ be a population of elements or objects under investigation. By assumption, L experts give predictions of values of unknown quantitative feature Y for objects $a \in \Gamma$, being already aware of their description $X(a)$. We assume that $X(a) = (X_1(a),...,X_j(a),...,X_n(a))$, where the set X may simultaneously contain qualitative and quantitative features X_j, $j = \overline{1,n}$. Let D_j be the domain of the feature X_j, $j = \overline{1,n}$, D_y be the domain of the feature Y. The feature space is given by the product set $D = \prod_{j=1}^{n} D_j$.

* The work was supported by the RFBR under Grant N07-01-00331a.

In this paper, we consider statements S^i, $i=\overline{1,M}$; represented as sentences of type "if $X(a)\in E^i$, then $Y(a)\in G^i$", where $E^i=\prod_{j=1}^n E_j^i$, $E_j^i\subseteq D_j$, $E_j^i=[\alpha_j^i,\beta_j^i]$ if X_j is a quantitative feature, E_j^i is a finite subset of feature values if X_j is a nominal feature, $G^i=[y_1^i,y_2^i]\subseteq D_y$. By assumption, each statement S^i has its own weight w^i. Such a value is like a measure of "assurance".

Preliminary Analysis

We begin with some definitions.

Denote by $E^{i_1 i_2}:=E^{i_1}\oplus E^{i_2}=\prod_{j=1}^n(E_j^{i_1}\oplus E_j^{i_2})$, where $E_j^{i_1}\oplus E_j^{i_2}$ is the *Cartesian join* of feature values $E_j^{i_1}$ and $E_j^{i_2}$ for feature X_j and is defined as follows. When X_j is a nominal feature, $E_j^{i_1}\oplus E_j^{i_2}$ is the union: $E_j^{i_1}\oplus E_j^{i_2}=E_j^{i_1}\bigcup E_j^{i_2}$. When X_j is a quantitative feature, $E_j^{i_1}\oplus E_j^{i_2}$ is a minimal closed interval such that $E_j^{i_1}\bigcup E_j^{i_2}\subseteq E_j^{i_1}\oplus E_j^{i_2}$ (see Fig. 1).

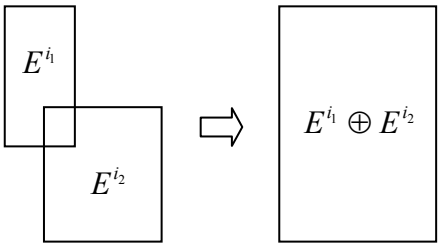


Fig. 1.

In the works [2, 3] we proposed a method to measure the distances between sets (e.g., E^1 and E^2) in heterogeneous feature space. Consider some modification of this method. By definition, put

$\rho(E^1,E^2)=\sum_{j=1}^n k_j\rho_j(E_j^1,E_j^2)$ or $\rho(E^1,E^2)=\sqrt{\sum_{j=1}^n k_j(\rho_j(E_j^1,E_j^2))^2}$, where $0\le k_j\le 1$, $\sum_{j=1}^n k_j=1$.

Values $\rho_j(E_j^1,E_j^2)$ are given by: $\rho_j(E_j^1,E_j^2)=\frac{|E_j^1\Delta E_j^2|}{|D_j|}$ if X_j is a nominal feature, $\rho_j(E_j^1,E_j^2)=\frac{r_j^{12}+\theta|E_j^1\Delta E_j^2|}{|D_j|}$ if X_j is a quantitative feature, where $r_j^{12}=\left|\frac{\alpha_j^1+\beta_j^1}{2}-\frac{\alpha_j^2+\beta_j^2}{2}\right|$. It can be proved that the triangle inequality is fulfilled if and only if $0\le\theta\le 1/2$.

The proposed measure ρ satisfies the requirements of distance there may be.

We first treat each expert's statements separately for rough analysis. Let us consider some special cases.

Case 1 ("coincidence"): $\max_j\max(\rho_j(E^{i_1},E^{i_1}\oplus E^{i_2}),\rho_j(E^{i_2},E^{i_1}\oplus E^{i_2}))<\delta$ and $\rho(G^{i_1},G^{i_2})<\varepsilon_1$, where δ, ε_1 are thresholds decided by the user, $i_1,i_2\in\{1,...,M\}$. In this case we unite statements S^{i_1} and S^{i_2} into resulting one: "if $X(a)\in E^{i_1}\oplus E^{i_2}$, then $Y(a)\in G^{i_1}\oplus G^{i_2}$".

Case 2 ("inclusion"): $\min(\max_j(\rho_j(E^{i_1}, E^{i_1} \oplus E^{i_2})), \max_j(\rho_j(E^{i_2}, E^{i_1} \oplus E^{i_2})) < \delta$ and $\rho(G^{i_1}, G^{i_2}) < \varepsilon_1$, where $i_1, i_2 \in \{1,...,M\}$. In this case we unite statements S^{i_1} and S^{i_2} too: "if $X(a) \in E^{i_1} \oplus E^{i_2}$, then $Y(a) \in G^{i_1} \oplus G^{i_2}$".

Case 3 ("contradiction"): $\max_j \max(\rho_j(E^{i_1}, E^{i_1} \oplus E^{i_2}), \rho_j(E^{i_2}, E^{i_1} \oplus E^{i_2})) < \delta$ and $\rho(G^{i_1}, G^{i_2}) > \varepsilon_2$, where ε_2 is a threshold decided by the user, $i_1, i_2 \in \{1,...,M\}$. In this case we exclude both statements S^{i_1} and S^{i_2} from the list of statements.

Consensus

Consider the list of l-th expert's statements after preliminary analysis $\Omega_1(l) = \{S^1(l),..., S^{m_l}(l)\}$. Denote by $\Omega_1 = \bigcap_{l=1}^{L} \Omega_1(l)$, $M_1 = |\Omega_1|$.

Determine values k_j from this reason: if far sets G^{i_1} and G^{i_2} corresponds to far sets $E_j^{i_1}$ and $E_j^{i_2}$, then the feature X_j is more "valuable" than another features, hence, value k_j is higher. We can use, for example, these values: $k_j = \frac{\tau_j}{\sum_{i=1}^{n} \tau_i}$, where $\tau_j = \sum_{u=1}^{M_1} \sum_{v=1}^{M_1} \rho(G^u, G^v) \rho_j(E_j^u, E_j^v)$, $j = \overline{1,n}$.

Denote by $r^{i_1 i_2} := d(E^{i_1 i_2}, E^{i_1} \bigcup E^{i_2})$.

The value $d(E,F)$ is defined as follows: $d(E,F) = \max_{E' \subseteq E \setminus F} \min_j \frac{k_j \mid E'_j \mid}{diam(E)}$, where E' is any subset such that its projection on subspace of quantitative features is a convex set (see Fig. 2), $diam(E) = \max_{x,y \in E} \rho(x,y)$.

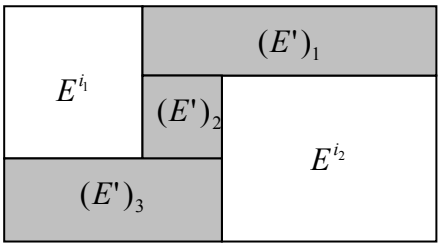


Fig. 2.

By definition, put $I_1 = \{\{1\},...,\{m_l\}\},...,\ I_q = \{\{i_1,...,i_q\} \mid r^{i_u i_v} \le \delta$ and $\rho(G^{i_u}, G^{i_v}) < \varepsilon_1 \quad \forall u, v = \overline{1,q}\}$, where δ, ε_1 are thresholds decided by the user, $q = \overline{2,Q}$; $Q \le M_1$. Let us remark that the requirement $r^{i_u i_v} \le \delta$ is like a criterion of "insignificance" of the set $E^{uv} \setminus (E^{i_u} \bigcup E^{i_v})$. Notice that someone can use another value d to determine value r, for example:

$$d(E,F,G) = \max_{E' \subseteq E \setminus (F \cup G)} \frac{\min(diam(F \oplus E') - diam(F), diam(G \oplus E') - diam(G))}{diam(E)}.$$

Further, take any set $J_q = \{i_1,...,i_q\}$ of indices such that $J_q \in I_q$ and $\forall \Delta = \overline{1, Q-q} \quad J_q \not\subset J_{q+\Delta}$ $\forall J_{q+\Delta} \in I_{q+\Delta}$. Now, we can aggregate the statements S^{i_1}, ..., S^{i_q} into the statement S^{J_q}:

S^{J_q} ="if $X(a) \in E^{J_q}$, then $Y(a) \in G^{J_q}$", where $E^{J_q} = E^{i_1} \oplus ... \oplus E^{i_q}$, $G^{J_q} = G^{i_1} \oplus ... \oplus G^{i_q}$.

By definition, put to the statement S^{J_q} the weight $w^{J_q} = \dfrac{\sum_{i \in J_q} c^{iJ_q} w^i}{\sum_{i \in J_q} c^{iJ_q}}$, where $c^{iJ_q} = 1 - \rho(E^i, E^{J_q})$.

The procedure of forming a consensus of single expert's statements consists in aggregating into statements S^{J_q} for all J_q under previous conditions, $q = \overline{1,Q}$.

Let us remark that if, for example, $k_1 < k_2$, then the sets E_1 and E_2 (see Fig. 3) are more suitable to be united (to be precise, the relative statements), then the sets F_1 and F_2 under the same another conditions.

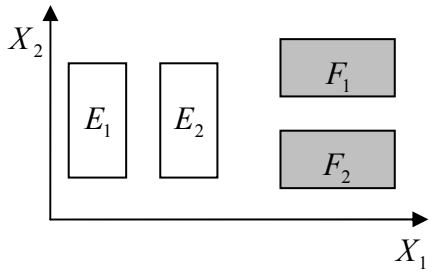


Fig. 3.

Note that we can consider another criterion of unification (instead of $r^{i_u i_v} \le \varepsilon$): aggregate statements S^{i_1}, ..., S^{i_q} into the statement S^{J_q} only if $w^{J_q} > \varepsilon'$, where ε' is a threshold decided by the user.

After coordinating each expert's statements separately, we can construct an agreement of several independent experts. The procedure is as above, except the weights: $w^{J_q} = \sum_{i \in J_q} c^{iJ_q} w^i$ (the more experts give similar statements, the more we trust in resulted statement).

Denote the list of statements after coordination by Ω_2, $M_2 := |\Omega_2|$.

Coordination

After constructing of a consensus of similar statements, we must form decision rule in the case of intersected non-similar statements. The procedure in such cases is as follows.

To each $h = \overline{2, M_2}$ consider statements $S^{(1)},...,S^{(h)} \in \Omega_2$ such that $\widetilde{E}^h := E^{(1)} \bigcap ... \bigcap E^{(h)} \neq \varnothing$, where $E^{(i)}$ are related sets to statements $S^{(i)}$.

Denote $I(l) = \left\{ i \middle| S^i(l) \in \Omega_1(l), \quad E^i(l) \bigcap \widetilde{E}^h \neq \varnothing \right\}$, where $E^i(l)$ are related sets to statements $S^i(l)$.

Consider related sets $G^i(l)$, where $l = \overline{1,L}$; $i \in I(l)$. Denote by $w^i(l)$ the weights of statements $S^i(l)$.

As above, unite sets $G^{(i_1)}(l_1),...,G^{(i_q)}(l_q)$ if $\rho(G^{i_u}, G^{i_v}) < \varepsilon_1 \ \ \forall u, v = \overline{1,q}$. Denote by $\widetilde{G}^1,...,\widetilde{G}^\lambda,...,\widetilde{G}^\Lambda$ the sets $G^i(l)$ after procedure of unification. Consider the statements $\widetilde{S}^\lambda$: "if $X(a) \in \widetilde{E}^h$, then $Y(a) \in \widetilde{G}^\lambda$".

In order to choose the best statement, we take into consideration these reasons:

1) similarities between sets $\widetilde{E}^h$ and $E^i(l)$;

2) similarities between sets $\widetilde{G}^\lambda$ and $G^i(l)$;

3) weights of statements $S^i(l)$;

4) we must distinguish cases when similar / contradictory statements produced by one or several experts.

We can use, for example, such values: $w^\lambda = \sum_{l=1}^{L} \frac{\sum_{i\in I(l)} (1-\rho(G^{(i)}(l), \widetilde{G}^{(\lambda)}))(1-\rho(E^{(i)}(l), \widetilde{E}^h)^2 w^i(l)}{\sum_{i\in I(l)} (1-\rho(E^{(i)}(l), \widetilde{E}^h)}$.

Denote by $\lambda^* := \arg\max_\lambda w^\lambda$.

Thus, we can make decision statement: $\widetilde{S}^h =$"if $X(a) \in \widetilde{E}^h$, then $Y(a) \in \widetilde{G}^{\lambda^*}$" with the weight $\widetilde{w}^h := w^{\lambda^*} - \max_{\lambda\neq\lambda^*} w^\lambda$.

Denote the list of such statements by Ω_3.

Final decision rule is formed from statements in Ω_2 and Ω_3. Notice that we can range resulted statements in Ω_2 and Ω_3 by their weights and exclude "ignorable" statements from decision rule.

Conclusion

Suggested method of forming of united decision rule can be used for coordination of several experts statements, and different decision rules obtained from learning samples and/or time series.

Bibliography

[1] G.Lbov, M.Gerasimov. Constructing of a Consensus of Several Experts Statements. In: Proc. of XII Int. Conf. "Knowledge-Dialogue-Solution", 2006, pp. 193-195.

[2] G.S.Lbov, M.K.Gerasimov. Determining of distance between logical statements in forecasting problems. In: Artificial Intelligence, 2'2004 [in Russian]. Institute of Artificial Intelligence, Ukraine.

[3] G.S.Lbov, V.B.Berikov. Decision functions stability in pattern recognition and heterogeneous data analysis [in Russian]. Institute of Mathematics, Novosibirsk, 2005.

Authors' Information

Gennadiy Lbov - *Institute of Mathematics, SB RAS, Koptyug St., bl.4, Novosibirsk, Novosibirsk State University, Russia; e-mail: lbov@math.nsc.ru*

Maxim Gerasimov - *Institute of Mathematics, SB RAS, Koptyug St., bl.4, Novosibirsk State University, Russia, e-mail: max_post@ngs.ru*

ENHANCING INFORMATION RETRIEVAL BY USING EVOLUTION STRATEGIES

Abdelmgeid Amin Aly

Abstract: *Similar to Genetic algorithm, Evolution strategy is a process of continuous reproduction, trial and selection. Each new generation is an improvement on the one that went before. This paper presents two different proposals based on the vector space model (VSM) as a traditional model in information Retrieval (TIR). The first uses evolution strategy (ES). The second uses the document centroid (DC) in query expansion technique. Then the results are compared; it was noticed that ES technique is more efficient than the other methods.*

1. Introduction

Since the 1940s the problem of Information Retrieval (IR) has attracted increasing attention, especially because of the dramatically growing availability of documents. IR is the process of determining relevant documents from a collection of documents, based on a query presented by the user.

There are many IR systems based on Boolean, vector, and probabilistic models. All of them use their model to describe documents, queries, and algorithms to compute relevance between user's query and documents.

Information Retrieval (IR) proposes solutions for searching, in a given set of objects, for those replying to a given description. IR tries to make a suitable use of these databases, allowing the users to access to the information which is really relevant in an appropriate time interval [1]. Unfortunately, commercial IR Systems (IRSs), usually based on the Boolean IR model [2], have provided unsatisfactory results. Vector space, probabilistic and fuzzy models, which have been developed to extend the Boolean model [3], as well as the application of knowledge-based techniques, have solved some of these problems, but there are still some lacks [4]. In the last few years, an increasing interest on the application of artificial intelligence (AI)-based techniques to IR has been shown with the aim of solving some of those lacks.

One of the AI areas with a considerable growth in the last decades is evolutionary computation (EC) [5], based on the use of models of evolutionary process for the design and implementation of computer-based problem solving systems. The different models which have been proposed within this philosophy are named in a generic way as evolutionary algorithms (EAs) [5].

In this paper, we introduce two different proposals using our IR model. One is using Evolution Strategies (ES) and the second using the Document Centroid (DC) in Query Expansion (QE) technique. Then the results are compared with our Traditional IR (TIR) model. To this end, we used the ES that presented the best performance, running it with two different fitness functions in the vector space model which is the most commonly used model in this type of application. We applied it to three well-known test collections (CISI, CACM and NPL). This allows us to generalize our earlier results and conclusion.

2. Antecedents

2.1. Evolutionary algorithms

EC [5] uses computational models of evolutionary processes as key elements in the design and implementation of computer-based problem solving systems. There is a variety of evolutionary computational models that have been proposed and studied, which are referred as EAs [5]. There have been four well defined EAs which have served as the basis for much of the activity in the field: Genetic Algorithms (GAs) [6], Evolution Strategies (ES) [7], Genetic Programming (GP) [8] and Evolutionary Programming (EP) [9].

An EA maintains a population of trial solutions, imposes random changes to these solutions, and incorporates selection to determine which ones are going to be maintained in future generations and which will be removed from the pool of trials. There are some important differences between the existing EAs. GAs [6] emphasize models of genetic operators as observed in nature, such as crossover (recombination) and mutation, and these re appllied to abstracted chromosomes with different representation schemes according to the problem being solved. Evolution strategies and evolutionary programming only apply to real-valued problems and emphasize mutational transformations that maintain the behavioral linkage between each parent and its off-spring.

As regards GP [8], it constitutes a variant of GAs, based on evolving structures encoding programs such as expression trees. Apart from adapting the crossover and mutation operators to deal with the specific coding scheme considered, the remaining algorithm components remain the same.

2.2. Evolution Strategies

Evolution strategies (ESs) were independently developed by Rechenberg [10], with selection, mutation, and a population of size one. Schwefel [11], introduced recombination and populations with more than one individual,

and provided a nice comparison of ESs with more traditional optimization techniques. Evolution strategies typically use real-valued vector representations. Evolution strategies are similar to genetic algorithms in that both attempt to find a (near-)optimal solution to a problem within a search space (all possible solutions to a problem) without exhaustively testing all solutions.

Evolution strategies are based on the principal of strong causality, which states that similar causes have similar effects. That is, a slight change to one encoding of a problem only slightly changes its optimality. The process of evolution strategy can be summarized by a relatively simple algorithm:

1. Generate some random individuals
2. Select the p best individuals based on some selection algorithm (fitness function)
3. Use these p individuals to generate c children (using mutation or recombination)
4. Go to step 2, until the ending condition is satisfied (i.e. little difference between generations, or maximum number of iterations completed).

2.3. Automatic Query Expansion

The automatic query expansion or modification based on term co-occurrence data has been studied for nearly three decades. The various methods proposed in the literature can be classified into the following four groups:

1. Simple use of co-occurrence data. The similarities between terms are first calculated based on the association hypothesis and then used to classify terms by setting a similarity threshold value [12], [13] and [14]. In this way, the set of index terms is subdivided into classes of similar terms. A query is then expanded by adding all the terms of the classes that contain query terms. It turns out that the idea of classifying terms into classes and treating the members of the same class as equivalent is too naive an approach to be useful [13], [15] and [16].
2. Use of document classification. Documents are first classified using a document classification algorithm. Infrequent terms found in a document class are considered similar and clustered in the same term class (thesaurus class) [17]. The indexing of documents and queries is enhanced either by replacing a term by a thesaurus class or by adding a thesaurus class to the index data. However, the retrieval effectiveness depends strongly on some parameters that are hard to determine [18]. Furthermore, commercial databases contain millions of documents and are highly dynamic. The number of documents is much larger than the number of terms in the database. Consequently, document classification is much more expensive and has to be done more often than the simple term classification mentioned in 1.
3. Use of syntactic context. The term relations are generated on the basis of linguistic knowledge and co-occurrence statistics [19], [20]. The method used grammar and a dictionary to extract for each term *t* a list of terms. This list consists of all the terms that modify *t*. The similarities between terms are then calculated by using these modifiers from the list. Subsequently, a query is expanded by adding those terms most similar to any of the query terms. This produces only slightly better results than using the original queries [19].
4. Use of relevance information. Relevance information is used to construct a global information structure, such as a pseudo thesaurus [21], [22] or a minimum spanning tree [23]. A query is expanded by means of this global information structure. The retrieval effectiveness of this method depends heavily on the user's relevance information. Moreover, the experiments in [23] did not yield a consistent performance improvement. On the other hand, the direct use of relevance information, by simply extracting terms from relevant documents, is proved to be effective in interactive information retrieval [24], [25]. However, this approach does not provide any help for queries without relevance information.

In addition to automatic query expansion, semi-automatic query expansion has also been studied [26], [27] and [28]. In contrast to the fully automated methods, the user is involved in the selection of additional search terms during the semi-automatic expansion process. In other words, a list of candidate terms is computed by means of one of the methods mentioned above and presented to the user who makes the final decision. Experiments with

semi-automatic query expansion, however, do not result in significant improvement of the retrieval effectiveness [26]. We use a document centroid (DC) as the basis of our query expansion.

3. System Framework

3.1. Building IR System

The proposed system is based on Vector Space Model (VSM) in which both documents and queries are represented as vectors. Firstly, to determine documents terms, we used the following procedure:

- Extraction of all the words from each document.
- Elimination of the stop-words from a stop-word list generated with the frequency dictionary of Kucera and Francis [29].
- Stemming the remaining words using the porter stemmer that is the most commonly used stemmer in English [3], [30].

After using this procedure, the final number of terms was 6385 for the CISI collection, 7126 for CACM and 7772 for NPL. After determining the terms that described all documents of the collection, we assigned the weights by using the formula (1) which proposed by Salton and Buckley [25]:

$$a_{ij} = \frac{\left(0.5 + 0.5\frac{tf_{ij}}{\max tf}\right) \times \log\frac{N}{n_i}}{\sqrt{\left(0.5 + 0.5\frac{tf_{ij}}{\max tf}\right)^2 \times \left(\log\frac{N}{n_i}\right)^2}} \quad (1)$$

where a_{ij} is the weight assigned to the term t_j in document D_i, tf_{ij} is the number of times that term t_j appears in document D_i, n_j is the number of documents indexed by the term t_j and finally, N is the total number of documents in the database.

Finally, we normalize the vectors, dividing them by their Euclidean norm. This is according to the study of Noreault et al. [31], of the best similarity measures which makes angle comparisons between vectors. We carry out a similar procedure with the collection of queries, thereby obtaining the normalized query vectors. Then, for applying ES, we apply the following steps:

- For each collection, each query is compared with all the documents, using the cosine similarity measure. This yields a list giving the similarities of each query with all documents of the collection.
- This list is ranked in decreasing order of similarity degree.
- Make a training data that consists of the top 15 document of the list with a corresponding query.
- Automatically, the keywords (terms) are retrieved from the training data and the terms which are used to form a query vector.
- Adapt the query vector using the ES approach.

3.2. The Evolution Strategy Approach

Once significant keywords are extracted from training data (relevant and irrevelent documents) including weights are assigned to the keywords. We have applied ES to get an optimal or near optimal query vector. Also we have compared the result of the ES approach with both the result obtained of (DC) and the traditional IR system. The (ES) approach will be explained in the following subsections.

Encoding & Fitness Functions

To implement an evolution strategy, the individuals in the population (solutions) need to be represented. Unlike genetic algorithms, which use bit strings, evolution strategies encode these individuals as vectors of real numbers (object parameters). Another vector of parameters, the strategy parameters, affects the mutation of the object parameters. Together, these two vectors constitute the individual's chromosome.

To distinguish whether one solution is more optimal than another, we use the cosine similarity as fitness function (2).

$$\frac{\sum_{i=1}^{t} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{t} x_i^{\,2} \cdot \sum_{i=1}^{t} y_i^{\,2}}} \qquad (2)$$

where X_i is the real representation weight of term i in the chromosome, Y_i is the real representation weight of that term in the query vector and t is the total number of terms in the query vector as in a given chromosome .

Forming the Next Generation

One key difference from genetic algorithms is that only the p most fit individuals in the population survive until the next generation (this form of selection is known as elitist selection). (Genetic algorithms usually use roulette wheel selection to give the fittest individuals a better chance of survival, but don't, like evolution strategies, guarantee that they will survive.) Using the fitness function as the evaluator, the p best individuals from the population are selected to be the parents of the next generation. A large value of p prevents bad characteristics from being filtered out of the gene pool (since they will persist from generation to generation), while a small value reduces variation in the gene pool, increasing the need for mutation. These p parent individuals produce a total of c children using mutation and recombination. The parents can be included in the next generation. Producing more children increases the probability of achieving better solutions, Mutation and recombination are used, but there are some Differences in how they are applied.

Mutation

To simulate mutation, random changes to the chromosome are made. These changes are necessary to add new genes to the gene pool; otherwise an optimal solution could not be reached if a necessary gene is absent.

Recombination

Recombination (also known as crossover) is the process where two or more parent chromosomes are combined to produce a child chromosome. Recombination is necessary in cases where each child is to have multiple parents, since mutation provides no mechanism for the "mixing" of chromosomes. We use a single point recombination, exchanges the weights of sub-vector between two chromosomes, which are candidate for this process.

Evolution Process

Figure 1 outlines a typical evolution strategy (ES). After initialization and evaluation, individuals are selected uniformly randomly to be parents. In the standard recombinative ES, pairs of parents produce children via recombination, which are further perturbed via mutation. Survival is deterministic and is implemented in one of two ways.

```
procedure ES; {
t = 0;
initialize population P(t);
evaluate P(t);
until (done) {
t = t + 1;
parent_selection P(t);
```

```
recombine P(t)
mutate P(t);
evaluate P(t);
survive P(t);
}        }
```

Fig. 1. The evolution strategy algorithm

The first allows the best children to survive and replaces the parents with these children. The second allows the N best children and parents to survive. Like EP, considerable effort has focused on adapting mutation as the algorithm runs by allowing each variable within an individual to have an adaptive mutation rate that is normally distributed with a zero expectation. Unlike EP, however, recombination does play an important role in evolution strategies, especially in adapting mutation.

3.3. The Query expansion approach

After using the vector space model (VSM) to represent the user's query and the documents. Each document d_k in the document database is represented by a document vector $\overline{d}_k$, the system calculates the degree of semilarity between the query vector $\overline{Q}$ and each document vector $\overline{d}_k$. Then, the system ranks the document according to their degrees of semiliarity with respect to the user's query from the largest to the smallest. Based on the relevant degree of relevant documents, get the top 15 documents for each query. The system considers each term appearing in any relevant document from the top 15 documents as a relevant term. The weight of each relevant term in each relevant document is calculated using formula (1). The average weight W_{avg} of each relevant term t_i is calculated as follows:

$$W_{avg} = \frac{\sum_{k=1}^{15} w_{ik}}{15} \tag{3}$$

where w_{ik} denotes the weight of relevant term t_i relevant document d_k. The result obtained from equation (3) represents the document Centroid (DC). The original and the additional terms together form the expanded query that is, consequently, used to retrieve documents, to get the result of the DC. We have three types of results, one for Traditional IR (TIR), second from adding the Document Centroid (DC), and third from the Evolution Strategies (ES).

4. Experimental Results

The test databases for our approaches are three well-known test collections, which are: the CISI collection (1460 documents on information science), the CACM collection (3204 documents on Communications), and finally the NPL collection (11,429 documents on electronic engineering). One of the principal reasons for choosing more than one test collection is to emphasize and generalize our results in all alternative test documents collections. The Experiments are applied on 100 queries chosen according to each query which does not retrieve 15 relevant documents for our IR system.

CACM Collection Results for 100 Queries

Table (1), and its corresponding graph represented in graph (1) both are using non-interpolated average Recall – Precision relationship. From this table we notice that ES gives a higher improvement than TIR with 21.35% and higher than DC with 39.6 % respectively as average values. the average number of terms of query vector before applying ES is 160.7 terms, these terms are reduced after applying ES to 16.83 terms, and increasing to 167.8 terms when applying DC approach.

Table (1): Shows the experimental results on CACM Collection

Average Recall-Precision Relationship			
Recall	Precision		
	TIR	DC	ES
0.1	0.72267	0.774552	0.81134
0.2	0.41646	0.52837	0.50888
0.3	0.36936	0.442366	0.45776
0.4	0.24673	0.267359	0.31126
0.5	0.21268	0.109873	0.2632
0.6	0.15801	0.005216	0.20032
0.7	0.14291	0.005216	0.17607
0.8	0.10728	0.005216	0.141
0.9	0.08965	0.005216	0.12236
Average	0.27397	0.238154	0.33247

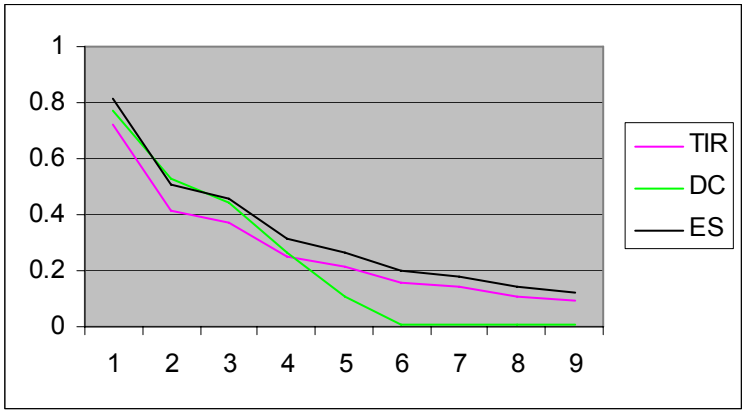


Fig. 2. Represents the relationship between average recall-precision for 100 queries on CACM

CISI Collection Results For 100 Queries

Table (2), and its corresponding graph represented in figure (3) both are using non-interpolated average Recall – Precision relationship. From this table we notice that ES gives a higher improvement than TIR with 17.66% and less than DC with 20.6% respectively as average values. The average number of terms of query vector before applying ES is 509.61 terms; these terms are reduced after applying ES to 358.84 terms, and increasing to 514.9 when applying DC approach.

Table (2): Shows the experimental results on CISI Collection

Average Recall-Precision Relationship			
Recall	Precision		
	TIR	DC	ES
0.1	0.67935	0.83819	0.84987
0.2	0.55781	0.753699	0.66449
0.3	0.46199	0.732035	0.5962

0.4	0.4007	0.67232	0.46771
0.5	0.34937	0.612855	0.40763
0.6	0.30394	0.454284	0.31125
0.7	0.25167	0.355882	0.27429
0.8	0.19887	0.187812	0.20741
0.9	0.14908	0.150724	0.16604
Average	0.37253	0.528645	0.43832

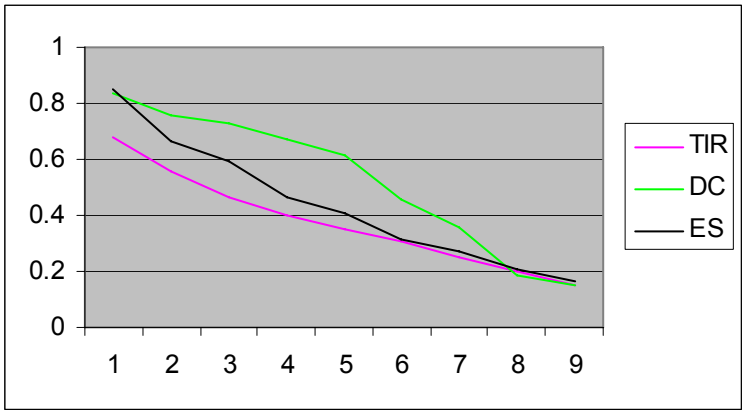


Fig. 3. Represents the relationship between average recall-precision for 100 queries on CISI

NPL Collection Results For 100 Queries:

Table (3), and its corresponding graph represented in graph (3) both are using non-interpolated average Recall – Precision relationship. From this table we notice that ES gives a higher improvement than TIR with 19.82% and higher than DC with 99.82% respectively as average values. The average number of terms of query vector before applying ES is 134.14 terms; these terms are reduced after applying ES to 16.8 terms, and increasing to 142.9 terms when applying DC approach.

Table (3): Shows the experimental results on NPL Collection

Average Recall-Precision Relationship			
Recall	Precision		
	TIR	DC	ES
0.1	0.73292	0.886421	0.80875
0.2	0.50337	0.457369	0.56654
0.3	0.43515	0.303652	0.50423
0.4	0.34047	0.147124	0.4184
0.5	0.31333	0.053247	0.39739
0.6	0.23999	0.00282	0.31045
0.7	0.21539	0.00282	0.27597
0.8	0.17428	0.00282	0.23302
0.9	0.14555	0.00282	0.20027
Average	0.34449	0.206566	0.41278

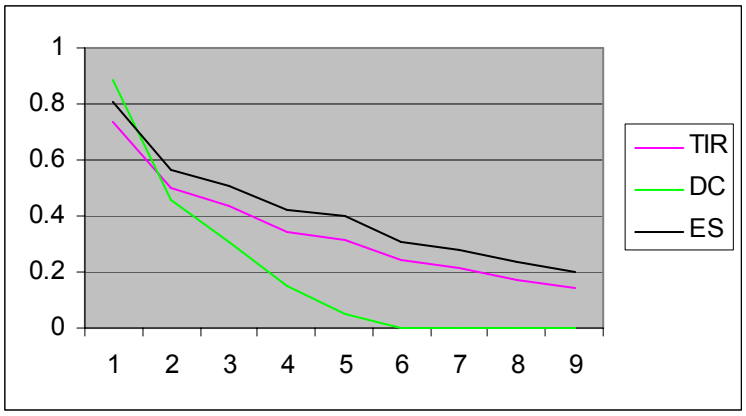


Fig. 4. Represents the relationship between average recall-precision for 100 queries on NPL

5. Conclusion

The goal is to retrieve most relevant documents with less number of non-relevant documents with respect to user's query in information retrieval system using evolution strategies. Our results have been applied on three well-known test collections (CISI, CACM and NPL), and compare the results of three variant methods (TIR, DC and ES). The results demonstrate that evolution strategies are effective optimization technique for Document retrieval.

Bibliography

[1] G. Salton, M.H. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

[2] C.J. Van Rijsbergen, Information Retrieval, second ed., Butterworth, 1979.

[3] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Adisson, 1999.

[4] H. Chen et al., "A machine learning approach to inductive query by examples: an experiment using relevance feedback", ID3, genetic algorithms, and simulated annealing, Journal of the American Society for Information Science 49 (8) (1998) 693–705.

[5] T. Bäck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, IOP Publishing and Oxford University Press, 1997.

[6] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1996.

[7] H.-P. GenerSchwefel, Evolution and Optimum Seeking, in: Sixth Generation Computer Technology Series, John Wiley and Sons, 1995.

[8] J. Koza, "Genetic Programming", On the Programming of Computers by means of Natural Selection, The MIT Press, 1992.

[9] D.B. Fogel, "System Identification trough Simulated Evolution: A Machine Learning Approach", Ginn Press, USA, 1991.

[10] I., Rechenberg, "Evolutions strategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution", Frommann-Holzboog, Stuttgart (1973).

[11] H.-P., Schwefel, Numerical Optimization of Computer Models, New York: John Wiley & Sons (1981).

[12] M.E., Lesk, "Word-word association in document retrieval systems", American Documentation, 20(1): 27-38, 1969.

[13] J.,Minker, Wilson, G.A., Zimmerman, B.H., "An evaluation of query expansion by the addition of clustered terms for a document retrieval system", Information Storage and Retrieval, 8(6): 329-48, 1972.

[14] K., Sparck-Jones, E.B., Barber, "What makes an automatic keyword classification effective?", Journal of the ASIS, 18: 166-175, 1971.

[15] H.J.,Peat, P., Willett, "The limitations of term co-occurrence data for query expansion in document retrieval systems", Journal of the ASIS, 42(5): 378-83, 1991.

[16] K., Sparck-Jones, "Notes and references on early classification work". SIGIR Forum, 25(1): 10-17, 1991.

[17] C.J., Crouch, "An approach to the automatic construction of global thesauri", Information Processing & Management, 26(5): 629-40, 1990.

[18] C.J.,Crouch, B.,Yong, "Experiments in automatic statistical thesaurus construction", SIGIR'92, 15th Int. ACM/SIGIR Conf. on R&D in Information Retrieval, Copenhagen, Denmark, 77-87, June 1992.

[19] G., Grefenstette, "Use of syntactic context to produce term association lists for retrieval", SIGIR'92, 15th Int. ACM/SIGIR Conf. on R&D in Information Retrieval, Copenhagen, Denmark, 89-97, June 1992.

[20] G.,Ruge, "Experiments on linguistically-based term associations", Information Processing & Management, 28(3): 317-32, 1992.

[21] G., Salton, "Experiments in automatic thesaurus construction for information retrieval", Information Processing 71, 1: 115-123, 1971.

[22] G., Salton, "Automatic term class construction using relevance-a summary of work in automatic pseudo classification", Information Processing & Management, 16(1): 1-15, 1980.

[23] A.F., Smeaton, C.J., van Rijsbergen, "The retrieval effects of query expansion on a feedback document retrieval system", The Computer Journal, 26(3): 239-46, 1983.

[24] Harman, D., "Relevance feedback revisited", SIGIR'92, 15th Int. ACM/SIGIR Conf. on R&D in Information Retrieval, Copenhagen, Denmark, 1-10, June 1992.

[25] G., Salton, C. Buckley, "Improving Retrieval Performance by Relevance Feedback", Journal of the ASIS, 41(4): 288-297, 1990.

[26] F.C., Ekmekcioglu, A.M., Robertson, P., Willett, "Effectiveness of query expansion in ranked-output document retrieval systems", Journal of Information Science, 18(2): 139-47, 1992.

[27] M., Hancock-Beaulieu, "Query expansion: advances in research in on-line catalogues", Journal of Information Science, 18(2): 99-103, 1992.

[28] S.J., Wade, P., Willett, " INSTRUCT: a teaching package for experimental methods in information retrieval". III. Browsing, clustering and query expansion, Program, 22(1): 44-61, 1988.

[29] H. Kucera , N. Francis. “Computational analysis of present-day American English”. Providence, RD: Brown University Press (1967).

[30] M. F. Porter. “An algorithm for suffix stripping. Program”, 14(3), 130–137 (1980).

[31] T. Noreault, M. McGill and M. B. Koll. “A performance evaluation of similarity measures, document term weighting schemes and representation in a Boolean environment”. Information retrieval research. London: Butterworths (1981).

Author's Information

A. A. Aly *– Computer Science Department, Minia University, El Minia, Egypt ; Email: abdelmgeid@yahoo.com*

KNOWLEDGE-BASED ROBOT CONTROL

Agris Nikitenko

Abstract: *The paper is related with the problem of developing autonomous intelligent robots for complex environments. In details it outlines a knowledge-based robot control architecture that combines several techniques in order to supply an ability to adapt and act autonomously in complex environments. The described architecture has been implemented as a robotic system that demonstrates its operation in dynamic environment.*

Although the robotic system demonstrates a certain level of autonomy, the experiments show that there are situation, in which the developed base architecture should be complemented with additional modules. The last few chapters of the paper describe the experimentation results and the current state of further research towards the developed architecture.

***Keywords**: Intelligent robots, autonomous intelligent systems, autonomous robots. Artificial intelligence.*

***ACM Keywords**: I.2.4.J – Representations, I.2.6.A – Analogies, I.2.6.E – Induction, I.2.8.G – Plan execution, formation, and generation, I.2.9.A – Autonomous Vehicles*

Introduction

Due to the constantly increasing interest about autonomous systems for application in various fields that require a certain degree of autonomy, it is necessary to develop platforms or architectures, which fit the demand. This paper describes an alternative knowledge-based architecture that combines several well known techniques of artificial intelligence in order to increase the system's autonomy. The most important advantage of the described architecture hides in the usage of the symbolic representation of the system's knowledge that is easy to use by the researcher and the system itself.

While the paper relies on the previous research in the field, only the most fundamental definitions are given [Nikitenko EMS2006, Nikitenko KDS 2005, Nikitenko 2005].

A complex environment is described with the following fundamental properties [Druzinin 1985]:

– uniqueness – usually complex systems are unique or number of similar systems is insignificant.

– hardly predictable – complex systems are very hard to predict.

– ability to maintain a certain progress resisting against some outer influence.

According to the sources used [Russell 2003, Huang 2003, Antsaklis 1996, Knapik 1997], an autonomous intelligent system is defined as any artificial intelligent system that can achieve its goals using its own knowledge, experience and available decision alternatives as well as operating without any outer assistance.

According to the definitions of the complex environment and autonomous system, the previous research resulted in development of knowledge-based architecture that supplies the basic functionality for autonomous system operating in a complex environment. The basic features of the developed architecture are outlined in the next section.

Basic Features of an Intelligent System

Summarizing the basic qualities of the proposed architecture are as follows [Nikitenko EMS 2006, Nikitenko KDS 2005, Nikitenko 2005]:

– Ability to reason about facts that are not observable directly by the system. This ability is achieved by means of the deductive reasoning. The proposed architecture does not state the kind of deductive reasoning that should be used. The only rule is that the selected deductive reasoning method has to address demands of a particular task. As it is described above the complex environments may be very dynamic and even with stochastic features. Therefore some uncertain reasoning techniques may be the most suitable. For example the experimental system implements a certainty factor based reasoning [Buchanan 1982].

– Ability to learn. As it is assumed above the intelligent system eventually will not have a complete model of the environment. Therefore the environment will be hardly predictable. Also complex environments are dynamic – in other words the system will face with new situations very often. Obviously, some adaptation mechanisms should be utilized. From point of view of intelligence the adaptation includes the following main capabilities: a capability of acquiring new knowledge and adjustment of the existing knowledge. The inductive reasoning module refers to the capability of learning. During an operation the intelligent system collects a set of facts (observations) through sensing the environment that forms an input for the learning mechanism.

– Ability to reason associatively. This feature is necessary due to the huge set of different possible situations that the intelligent system may face with in the complex environments. For example, there may be two different situations that can be described by n parameters (n is big enough number) where only k parameters are different (k is small enough number). Obviously these situations may be assumed as similar. Therefore an associative

reasoning is used – to reason about situations that are observed for the first time by the intelligent system similarly to reasoning about experienced situations. The associative reasoning is realized through using associative links among similar situations (descriptions). Each situation may be accessed or identified by a set of features thus this mechanism operates in similar manner to the associative memory [Kokinov 1988, Wichert 2000]. An issue about which situations should be linked is conditioned by a particular task or goals of the system's designer.

– Ability to sense an environment. This feature is essential for any intelligent system that is built to be more or less autonomous. This feature also includes an ability to recognize situations that the system has faced with as well as an ability to obtain data about unknown situations. All sensed data is structured in frames (see below). During the frame formation process the sensed environment's state is combined with system's inner state thereby allowing the system to reason about the system itself. Also the sensed system's and environment's states are used to realize a feedback in order to adjust the system's knowledge. Thereby the system's flexibility is increased.

– Ability to act. This feature is essential for any intelligent system that is designed to do something. If the system (autonomous) is unable to act, it will not be able to achieve its goals. The way of acting and the purpose of acting vary depending on the goals of the system's designer.

The listed above features form a basis for an intelligent system that operates in a sophisticated environment. According to the features of complex systems that are listed above, any of them may be implemented as it is needed for a particular task. In other words the implementation methods and approaches are dependant on the purposes of the system itself. Nevertheless the main question is how to bind all of them in one whole - one intelligent system. Obviously, there is a necessity for some kind of integration. There are many good examples of different kinds of integration that may be found in widely available literature devoted to hybrid intelligent systems [Goonatilake 1995].

The developed architecture is based on so called intercommunicating hybrid architecture where each of the integrated modules is independent, self-contained, intelligent processing unit that exchanges information and performs separate functions to generate solutions [Goonatilake 1995]. The developed architecture in more details is described in the next section

Architecture of the Intelligent System

According to the list of the basic features there can be outlined the basic modules that correspond to the related reasoning techniques. The modules are outlined in the figure 1.

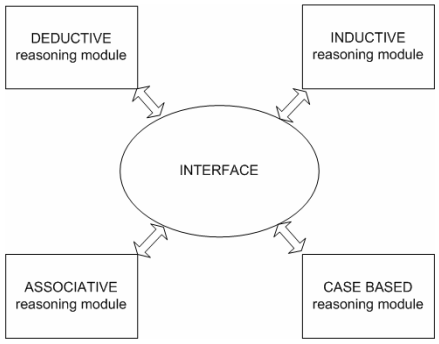


Figure 1. Basic modules

According to the figure 1, there are four basic modules that form system's kernel. The modules fulfil the following basic functions

– Deductive reasoning module This module performs deductive reasoning using if..then style rules [Luger 2002, Russell 2003]. In order to implement the adaptation functionality, this module may exploit some particular uncertain reasoning technique. In the proposed architecture the main purpose of this module is to predict

(forecast) future states of the environment as well as the inner state of the system. During the reasoning process if..then rules are used in a combination with the input data obtained from the sensors.

– Inductive reasoning module This module performs an inductive reasoning. It learns new rules and adds them to the rule base. Again the proposed architecture does not state what kind of inductive learning technique is used. The only limitation is the requirement to produce rules that could be used by the deductive reasoning module. For example, if the fuzzy reasoning is used, then the result is a set of fuzzy rules.
– Case based reasoning module Case based reasoning operates with "best practice" information that helps to reduce planning time as well as provides this information to modeler in an explicit manner.
– Associative reasoning module. This module links situations according to the similarity among them thus allowing to reason associatively. In the robotic system the similarity measure is calculated using the following formula:

$$yi = \begin{cases} 1, if \sum_{j=1}^{n} \partial(x_j) \geq T \\ 0, if \sum_{j=1}^{n} \partial(x_j) < T \end{cases} \quad (1)$$

In the formula (1):

yi – 1 – if the i-th situation is similar to the given;

n – a number of attributes that describes each situation;

xj – value of the j-th attribute of the given situation;

T – a threshold value – an number of attributes which values are equal for the given and the i-th situation. In the robotic system (see below) is applied only for 8 IR sensors, all other situation parameters have to be equal. Therefore the T's value may be changed from 1 to 8.

$\partial(x)$ - 1 if the value of the j-th attribute is equal for both situations;

This module considerably reduces the overall amount of knowledge processed by the system because there is no need to store every experienced situation just unique situations, which are not like any other experienced before.

Of course, the intelligent system needs additional modules that would supply it with the necessary information about the environment and mechanisms to perform some actions. Therefore the basic architecture shown in figure 1 is complemented with few additional modules. The enhanced architecture is depicted in the figure 2. The additional modules (drawn in grey) have the following basic functions:

– Planner module. This module is one of the central elements of the system. Its main function is to plan future actions that lead to achievement of the system's goals. During the planning process three of the basic reasoning techniques are involved – deductive, case based and associative reasoning. A result of the planner is a sequence of actions that are expected to be accomplished by the system thereby achieving its goals.

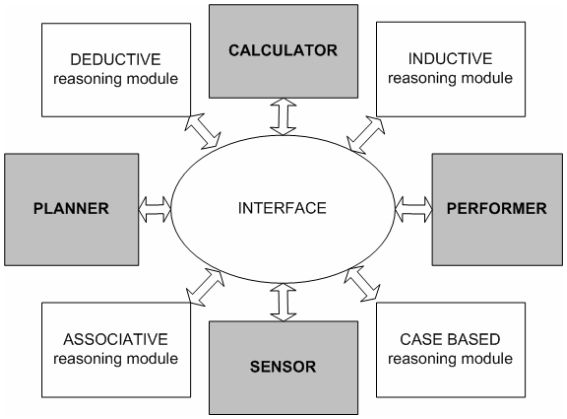


Figure 2. Enhanced architecture

– Sensor module. The module's purpose is to collect information from the sensors about the environment's and the system's states. The sensed information is portioned in separate frames (see below) and forwarded to the interface (see figure 3). Once the information is forwarded, it is available for the other modules.

– Performer module. This module performs a sequence of actions that are included in the plan. Also this module uses information about the system's and the environment's current states in order to determine whether the instant actions can be accomplished.

– Calculator module. This module collects and produces any reasoning relevant quantitative data. For example, in the robotic system (see below) this module is used to calculate certainties of the rules including those rules that are newly generated by the inductive reasoning module. Functionality of the module may be enhanced according to the necessities of the particular tasks or goals of the system's designer.

As it is depicted in figures 1 and 2, all of the modules use the central element – Interface in order to communicate to each other. They are not communicating to each other directly thereby a number of communication links is reduced as well as all of the information circulating in the system is available for any module. A architecture of the interface is depicted in the following figure 3:

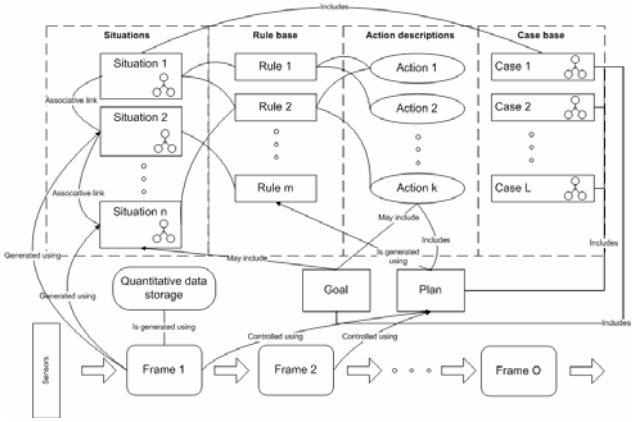


Figure 3 Architecture of the interface.

The architecture consists of several basic elements. The fundamental element of the whole architecture is situation.

Situations. Situations are the key elements in the interface structure. They correspond to the situations which are experienced by the system. Every situation is described with a set of features (attributes). Each attribute is described with its value. As it is depicted in the figure 3 situations are linked to each other by associative links. These links form the basis for associative reasoning. When the intelligent system runs into a certain case the most likely situation is activated. It is used for reference to rules and cases. If there is no rule that can be triggered, then the system tries to trigger rules that refer to the bounded similar situations (with same degree of likeness). The result may be less feasible, but using association among situation the system can run out of the dead end cases. The idea is obtained from the associative memory mechanism [Kokinov 1988, Wichert 2000]. It also reduces the impact of the sensor errors on the reasoning process, because allows to use similar not directly matching situation. Thereby the overall amount of knowledge is reduced as well.

Rules. Rules are any kind of notation that represents causalities. In the practical experimentations a well known if..then notation was used [Luger 2002, Russell 2003]. As it is depicted in the figure 3 rules are linked to situations and actions. When the system activates several situations by using associative links the appropriate rules are also activated thus system can scan a set of "associated" rules as well. This simple mechanism improves the ability to adapt. As it is depicted in the figure 3 rules are linked to actions. Thereby rules through the deductive reasoning are used in planning process. In the robotic system each rule is complemented with a certainty factor which is used during the reasoning process (the certainty theory's simplified practical model is used) [Buchanan 1982]. The certainty factor is calculated using the following formula:

$$CF(Rule) = \frac{S}{N} \tag{2}$$

In the formula (2):

CF(Rule) – certainty factor of the rule;

S – number of times when the rule's forecasted values of the situation attributes (one or many) were observed by the system;

N – number of times when the rule has been used;

Case quality or value is calculated using the same formula. In that case:

S – number of times when case was used and appropriate goal was achieved;

N – number of times when the case was used;

Actions. Actions are symbolic representations that can be translated by the intelligent system and cause the system to do something. For example "turn to the right" causes the system to turn to the right by 900. Each action consists of three parts: precondition, body and postcondition. Precondition is every fact (attribute of the situation) that should be true before the action is executed. For example, before opening the door it has to be unlocked. Body is a sequence of basic (or lover level) actions that are executed directly – for example a binary code that forwarded to a motor controller causes the motors to turn (in the case of robotic system). Post conditions are factors that will be true after the execution. For example after opening the door, the door is opened.

Cases. Cases are direct descriptions of the system's experience. Mathematically a case is described as follows:

$$Case = \{E, Pl, G\} \tag{3}$$

In the formula (3):

Case – The case;

E – situation or input;

Pl – plan which leads to achievement of the goal;

G – goal;

In the robotic system each case was complemented with reliability factor that is calculated using the formula (2), with difference that S is a number times when the case has been used and the goal was achieved.

Frames. Frames are data structures that contain the sense array from the environment and the system. It means that frames contain snapshots of the environment's and the system's states. Mathematically it may be described as follows:

$$Frame = \{En, Sy\} \tag{4}$$

In the formula (4):

En – a snapshot of the environment's state;

Sy – a snapshot of the system's state;

As it is depicted in figure 3 frames are chained one after another thus forming a historical sequence of the environment's and the system's states. Frames form an input data for the learning (induction module) algorithms as well.

Goal. The goal is a task that has to be accomplished by the system. It can be defined in three different ways: as a sequence of actions that should be done, as some particular state that should be achieved or as a combination of the actions and the states. The third option is implemented in the robotic system described below. Thereby the goal is described as:

$$G = \{S, M, C\} \tag{5}$$

In the formula (5)

G – the goal;

S – a set of states that has to be achieved;

M – a set of actions that has to be performed;

C – order constraints that order elements of the sets S and M;

Plan. Plan is a sequence of actions that is currently executed by the system. It may be formed using both basic and complex actions. After the plan is accomplished it is evaluated depending on whether the goal is achieved or not thereby forming feedback information for the calculator module.

Quantitative data. This element is used to maintain any kind of quantitative data that is produced by the calculator module and is used during the reasoning process. For example it may contain certainties about facts or rules, possibilities etc. Quantitative data is collected during the reasoning process as well as during the analysis of the input data - feedback data. All of those components together form an interface for the basic modules: Inductive, Deductive, Case based and Associative reasoning. The architecture is implemented as experimental robotic system that is shortly described below.

Experimental Robotic System

The robotic system is an autonomous intelligent system that encapsulates all of the mentioned above elements of the proposed architecture and interface among them. The system's input consists of the following sensors:

– Eight IR (infrared) range measuring sensors;
– Electronic compass;
– Four bump sensors (two front and two rear micro switches)
– Four driving wheel movement measuring resistors (two for each driving wheel in order to achieve reliable enough measurements).

The sensors and robotic system is depicted in the figure 5.

Two Basix-X [BasicX] microprocessors are used in order to communicate with PC and to perform input data preprocessing and formatting. Prepared and formatted data as frames (see above) are sent to the PC via RS-232 connection [Strangio 2005]. Few screenshots of the controlling software are shown in the following figures:

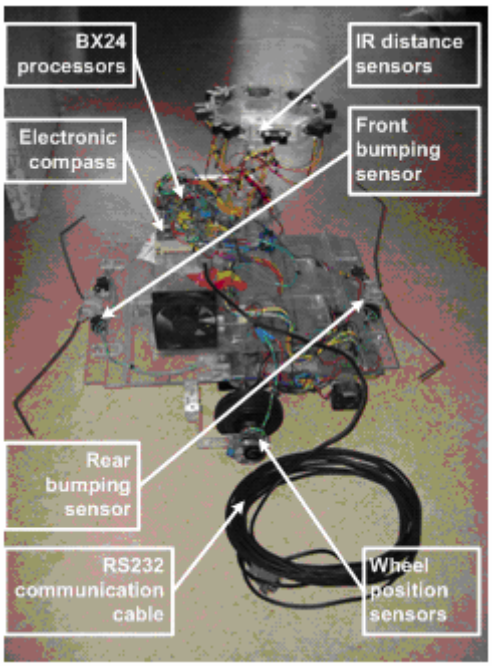


Figure 5. The robotic system.

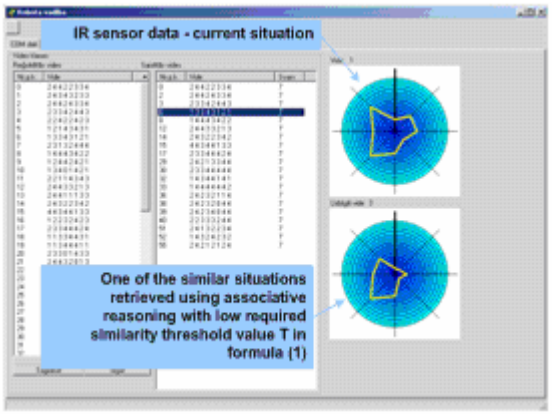


Figure 6. Contr. Software – sensor data section.

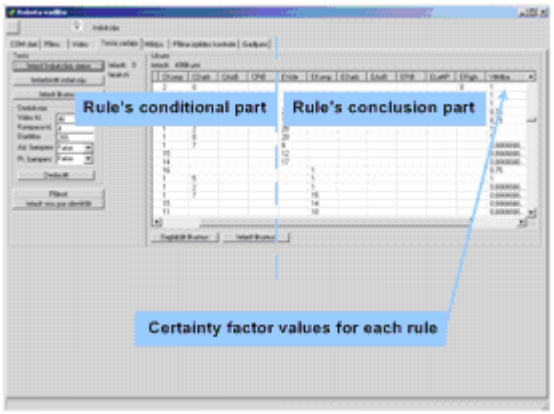


Figure 7. Contr. Software – rule base section.

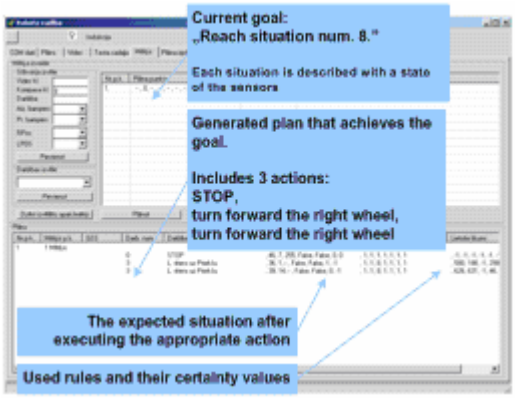


Figure 8. Contr. Software – goal definition

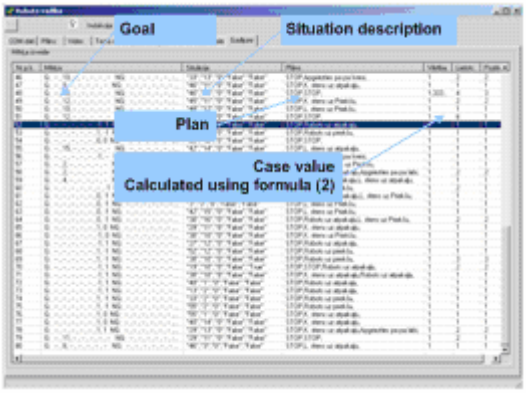


Figure 9. Contr. Software – Case base section.

All other modules of the intelligent system are implemented as a PC-based software that has a user-friendly interface allowing a simple following the system's operation, collection of the research sensitive data, changing system's goals etc.…

The system is built for research purposes only. In other words, it is built for experiments in order to examine and validate the proposed architecture. Therefore the system's user interface is built to be as flexible as possible allowing its user to manipulate with the robot's state, goals and results at the runtime. The most important features of the robotic system are:

- Ability to work with multiple goals with mixed structure that may include – actions, states or both;
- Ability to adapt via using inductive learning algorithm C 4.5[Quinlan 1996]. Even with the well known disadvantages of the axis parallel classification that is used in the algorithm, the system demonstrates acceptable adaptation abilities. In practical implementations there may be used other methods described in [Centu-Paz 2000, Pappa 2004] that eliminate these problems.
- Case-Based reasoning is used to store information about best-practice cases and to use this information during the planning process.
- Ability to reason using Certainty theory ideas thus allowing addition of new rules that may be conflicting with the existing ones in the rule base.
- Ability to reason using associative links among the situations.
- System's knowledge and the system's state relevant data is stored and processed in explicit and easy to follow manner thus demonstrating the advantages of the used knowledge based techniques.

It is important to stress that at the very beginning of the system's operation it has no information about the consequences of each action – it needs to learn them. Thereby the bottom-up learning is used. But if it is necessary the system's rule base may be filled with rules, cases and other research relevant information thus allowing to model particular state of the system.

Experiments

There are several experiments accomplished by the author in order to examine the system and its behaviour in different working conditions. One of the most important experiments is described in this section.

Experiment's goals

The experiment's primary goal is to prove the architecture's ability to adapt and autonomously achieve the given goals as well as to characterize the system's behaviour in uncertain conditions. A secondary goal is a demonstration of the system's operation in uncertain and dynamic conditions.

Experiment conditions

In order to meet the experimentation goals, a 3 × 3 m arena is used. The robotic system can freely move around the arena, but cannot move outside the arena because of special 50 cm high walls around it.

At the beginning of the experiment the system's knowledge base is empty containing only a list of actions that may be executed. It means that the system has to acquire knowledge form the observations (sequence of sensor state snapshots) in order to achieve the given goals. 120 random goals one after another are given to the system. The maximum plan length is limited to 3 actions.

In order to simulate uncertain conditions, the system is randomly turned in unexpected directions thus causing random state transitions. In order to simulate random events, the experimenter from time to time is walking inside the arena in unexpected directions thus causing random state transitions.

Results

The main measure is effectiveness that is calculated using the following formula:

$$E = \frac{M}{P} \times 100\% \tag{6}$$

In the formula (6):

E – effectiveness expressed in percents; M number of goals achieved by the system; P – number of plans generated during the achievement of the goals;

The results are outlined in the following graphics:

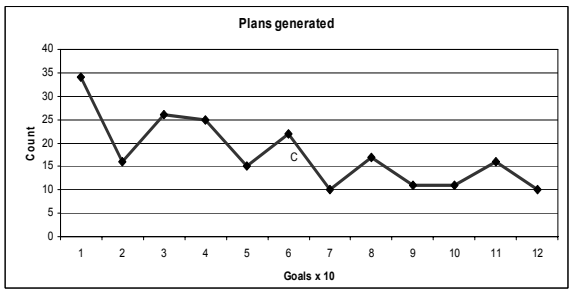


Figure 10. A number of plans constructed by the system

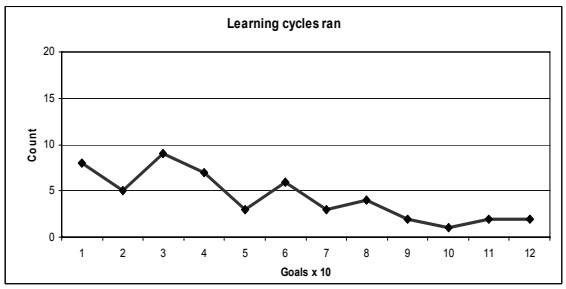


Figure 11. A number of learning cycles

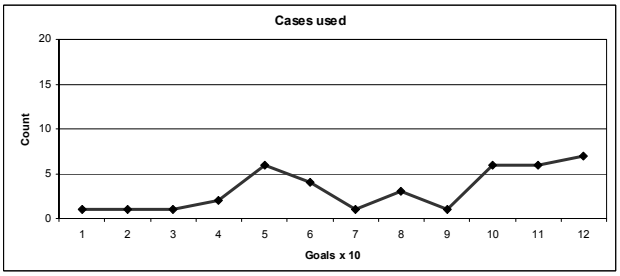


Figure 12. A number of cases used to construct plans

The first graphic (see figure 10) shows the number of plans constructed by the system during the run. As it is depicted in the figure 4 at the beginning the number of plans is very high (34), approximately 3.4 plans for one goal. While the system adapts the number of constructed plans is gently decreasing until reaches the minimum – 10 plans for 10 goals. The second graphic (see figure 11) shows the number of learning cycles fired by the system. The shape of the graphic almost repeats the first graphic. The learning mechanism at first collects a set of examples which is used for rule generation. Thereby the less actions are performed the more time is required for collection of the set with appropriate amount of examples. The third graphic shows the number of cases used for plan acquisition (each case includes a ready-to-use plan). The number of cases used is increasing in correspondence with the growth of the system's experience. The last 10 goals are achieved using 7 cases. The planning took almost 6 s during the achievement of the last 10 goals while the retrieval of the appropriate case

took only 0.05 s (more than x100 faster). This emphasizes the importance of the case based reasoning in autonomous systems. Also results of the other experiments showed that associative reasoning more the 10 times reduced the amount of knowledge (situation descriptions, rules and cases) used by the system. This lets to decrease planning time and increase flexibility of the system.

Other experiments

The systems behaviour is compared with system's operation with randomly generated plans. During this experiment remarkably simpler goals are given – goals that may be achieved with one action. For example, a certain state when one wheel is turning forward (with empty knowledge base the system does not 'know' consequences of any action, they has to be learned). With each plan length (1, 2, 3, 4, 6) 10 goals are given for achieving in the same arena.

The results are shown in the following figure:

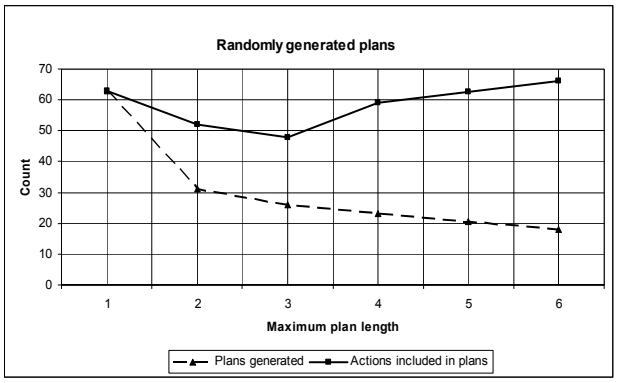


Figure 13. Number of generated plans and included actions

This experiment shows that the maximum effectiveness (aprox. 2 plans per goal) is reached with plan length 6 while it is necessary only one action. The figure 10 shows that at the end of experiment for each goal only 1 plan is required in more complex conditions. Consequentially, the used mechanisms and the knowledge representation schema provide means for certain convergence shown in the previous figures.

Another experiment is conducted in order to examine an importance of the associative reasoning. Like in the previous experiment with random plans, the same conditions are used as well as 10 goals with each example set are given.

In the following figure is shown system's performance with different learning sets – 10, 15, 20 and 25 examples per set:

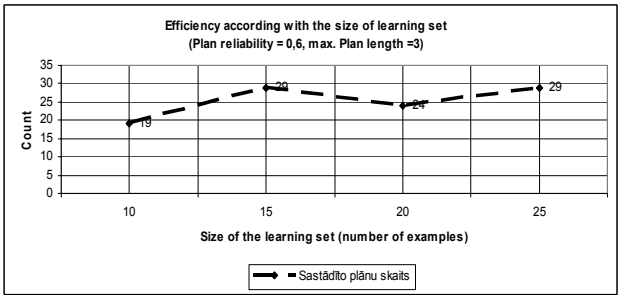


Figure 14. Number of generated plans with different learning sets

During the next run an associative reasoning is used with different threshold values T (see formula (1) and appropriate descriptions). All other conditions remain the same. Results are shown in the following figure:

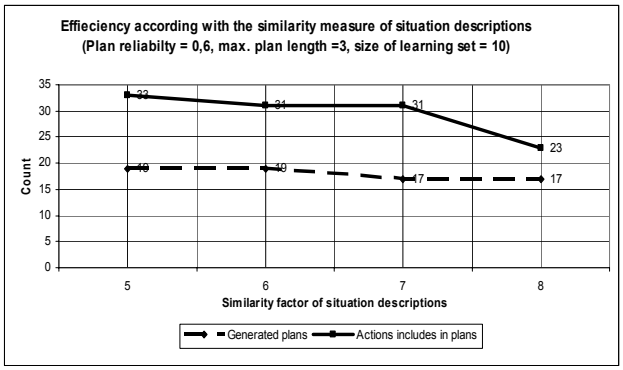


Figure 15. Number of generated plans and included actions with different T values

In both runs a considerable performance growth is achieved comparing with the random plan generation results. When the associative reasoning is used the system's performance is slightly higher. The slight difference is caused by the small number of goals achieved because the associative reasoning is based on the previous experience. Therefore if the system is not experienced enough the associative reasoning cannot provide considerable performance growth. Nevertheless the performance is increased.

Importance of the case based reasoning may be observed in figure 12, where the usage of cases is increasing together with overall experience of the system. For example, the first 10 goals are reached without using cases while the last 10 goals are achieved applying 7 cases.

Other experiments that are not presented in this paper were conducted in order to better describe some specific parameters of the system's behavior under different conditions.

Current State of Further Research

As it is shown above, the implemented architecture as a basic input unit uses frame, which consists of system's and the environment's state snapshots. Each snapshot is described by means of system's sensor states. Thereby the goal also has to be described in means of system's senor states. This results in a significant limitation if the system has to achieve some global goal – goal that is "out of the system's range of sight". For better description let us consider the following simple task:

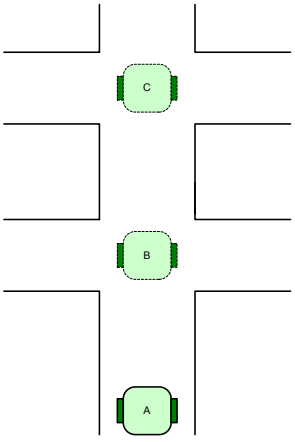


Figure 15. System's task

Where, A – the systems' current state, C – system's goal state and B – unwanted state.

If the desired state is C and it is out of the system's sensor range, then it is more likely that the system will stop at the state B because its representation by means of the system's sensor state is identical with the state's C representation. In other words, the system can achieve only local goals. However, having information about states B and C, it is possible to define two separate goals that follow one after another. Obviously, the system might use a geographical or topological map of the environment that could provide the necessary information

about states B and C. That is the main direction of the further research that is currently undertaken ie. enhancement of the developed architecture in order to use geographical and topological maps. Map as an additional model would allow to operate with tasks that require: environment investigation and construction of the map, self localization, object lookup and others.

Currently the research is based on a computer model of the robotic system and its environment. The computer model in contradiction with the real system allows much easer to compare different techniques used for implementation of certain modules. The model uses MS Robotic Studio infrastructure and visualization tool while the robot control software is implemented using a general purpose programming language. The used infrastructure provides all basic functionality for modeling a robot and its environment as a visual entity and as a physical entity as well. The following figures show the difference between mentioned two views of the same modeled entity.

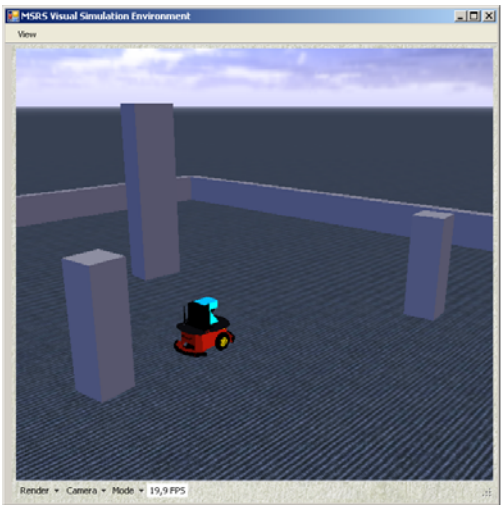


Figure 16. Model's visual view

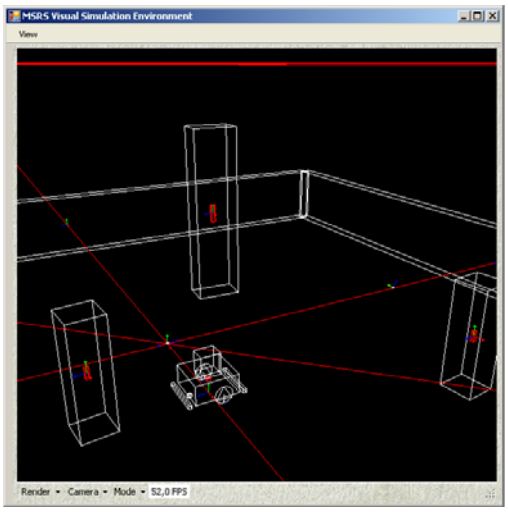


Figure 17. Model's physical view

The visual model is used for convenience of the user, but the physical view is used for calculations. The physical view is less detailed in order to maintain only the essential calculations of the modeled entities.

The modeling tool has been chosen because of the following features:

- Functionality, which is tailored for specific needs of robotics ie. all of the most popular devices are supported (the others may be added by the user), wheel traction, mass and inertia and other very important physical aspects are supported.
- Enhanced service oriented architecture that supports event-driven control over the modeled robotic system
- Platform independence of the controlling software that allows to switch between model and real robot by adjustment of the service references without any changes in the controlling code.
- Support of serialization of the developed model that allows to "save" the current state and continue modeling from this particular state whenever it is necessary.

The mentioned basic features make the MS Robotic Studio very convenient tool for development and modeling of robotic systems.

Currently all of the basic modules are already implemented using the modeling tool. The link between core modules and map module is in the development stage. Although the system is not developed yet, the test drives of the implemented modules show the ease of use of the mentioned modeling tool and performance of the development that is increased due to the usage of the high level programming language instead of the low level processor specific languages.

As it is stressed in author's previous papers it is necessary to examine the developed architecture under different configurations. For example, the architecture might be implemented using different induction algorithms. The

developed model will allow to switch among different possible configurations for more detailed examinations of the developed architecture in the same conditions.

Conclusions

The conducted experiments show that in spite of the known drawbacks of the used algorithms (in particular C4.5) the described architecture provides means for convergence in uncertain and dynamic conditions. However a deeper analysis outlines important limitations that may be avoided via an appropriate goal definition or via using additional models. The further research activities are concentrated on geographical and topological map integration into the developed architecture as a separate module. That would provide the necessary information for achievement of goals that are “out of the direct sight” of the robot’s sensors and widen the sphere of application of the architecture.

The conducted experiments do not compare performance of different configurations of the developed architecture that is still an important issue of further examination.

Bibliography

[Antsaklis 1996] P.J. Antsaklis, M. Lemmon, J.A. Stiver, Learning to be autonomous: Intelligent supervisory control. Intelligent Control Systems: Theory and Applications. England, IEEE Press, 1996, pp. 28 – 62.

[BasicX] © NetMedia Inc. www.basicx.com

[Buchanan 1982] B.G.Buchanan, R.O.Duda, Principles of Rule-Based Expert Systems. Stanfrod, USA, Stanford University, 1982, 62 p., STAN-CS-82-926

[Burkhard 2001] H.Burkhard, J.Bach, R.Berger, B.Brunswieck, M. Gollin, Mental models for Robot Control, Proceeding of International seminar “Advances in Plan – based control of Robotic agents”, Dagstuhl Castle, Germany , 2001.

[Centu-Paz 2000] E.Cantú-Paz, C.Kamath, Using Evolutionary Algorithms to Induce Oblique Decision Trees. Proceedings of Genetic and Evolutionary Computation Conf. (GECCO), Las Vegas, USA, 2000.

[Davidsson 2000] P.Davidsson, Multi Agent Based Simulation: Beyond social simulation, In Multi Agent Based Simulation, LNC series 1979, Springer Verlag, 2000, pp. 97 - 107.

[Drogoul 2002] A.Drogoul, D.Vanbergue, T.Meurisse, Multi-Agent Based Simulation: Where are the Agents?. Proceedings of Multi-Agent Based Simulation (MABS), Bologna, Italy, 2002.

[Druzinin 1985] V.V.Druzinin, D.S. Kontorov “Systemtechique”, Radio and Communication, 1985, 200 p., (В.В. Дружинин, Д.С. Конторов «Системотехника» Радио и Связь 1985, 200 с.)

[Goonatilake 1995] S.Goonatilake, S.Khebbal, Intelligent hybrid systems. Biffind Lane, Chichester, West Sussex, England, John Wiley & sons Ltd. 1995, 325 p., ISBN 0-471-94242-1

[Huang 2003] H.Huang, E.Messina, J.Albus, Autonomy Level Specification for Intelligent Autonomous Vehicles: Interim Progress Report. Proceedings of the 2003 Performance Metrics for Intelligent Systems Workshop, Gaithesburg, USA, August 16 - 18, 2003, pp. 1 – 7.

[Knapik 1997] M.Knapik, J.Johnson, Developing Intelligent Agents for Distributed Systems. USA, Osborne/McGraw-Hill, 1997, 398.p. ISBN 0-0703-5011-6.

[Kokinov 1988] B. Kokinov, Associative memory-based reasoning: How to represent and retrieve cases. In T. O'Shea and V. Sgurev (Eds.), Artificial intelligence III: Methodology, systems, applications. Amsterdam, The Netherlands, Elsevier Science Publishers B.V., 1988, pp. 55 – 58.

[Luger 2002] G.F.Luger, Artificial intelligence – Structures and Strategies for Complex problem Solving. 4th edition, Addison Weslwy, 2002, translation to rus., Искуственный интеллект – Стратегии и методы решения сложных проблем, Москва, Россия, Издательский дом «Вильямс», 2005, 863.с. ISBN 0- 201-64866-0.

[Nikitenko EMS 2006] A.Nikitenko, Autonomous Intelligent Agent Control in Complex Environments, Barcelona, Spain, EMSS 2006 Conference Proceedings, 2006. p. 251 - 260

[Nikitenko KDS 2005] A.Nikitenko, Robot Control Using Inductive, Deductive and Case Based Reasoning. Varna, Bulgaria, KDS 2005 conference proceedings, 2005, Vol. 2, pp. 418 - 427.

[Nikitenko 2005] A.Nikitenko, Intelligent Agent Control Using Inductive, Dedudctive and Case Based Reasoning. Riga, Latvia, ECMS 2005 Conference Proceedings, 2005, pp.486 - 492.

[Pappa 2004] G.L.Pappa, A.A.Freitas, Towards a genetic programming algorithm for automatically evolving rule induction algorithms. Proceedings of ECML/PKDD-2004 Workshop on Advances in Inductive Learning, Pisa, Italy, 2004, pp. 93 – 108.

[Quinlan 1996] J.R.Quinlan, Improved Use of Continuous Attributes in C4.5. In Journal of Artificial Intelligence Research, 1996, Vol. 4, pp. 77 – 90

[Russell 2003] S.Russell, P.Norvig, Artificieal Intelligence – A Modern Approach. 2nd edition, Upper Saddle River, New Jersey, USA, Pearson Educations Inc., 2003, 1080 p., ISBN 0-13-080392-2.

[Strangio 2005] C.E.Strangio, The RS232 standard [online], CAMI Research Inc., Lexington, Massachusetts, 1993, [cited: 20.12.2005]. Available from World Wide Web: <http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html>.

[Wichert 2000] A.M.Wichert, Associative Computation, Ph.D. thesis, Ulm, Germany, Univeristy of Ulm, July 2000.

[Wooldrige 1995] M.Wooldrige, N.Jennings, Intelligent Agents: Theory and practice. Knowledge Engineering Review, Cambridge, England, Cambridge University Press, 1995, Vol. 10, No 2, pp. 115 – 152.

Author's Information

Agris Nikitenko - *Riga Technical University, Department of System Theory, Meza street 1 k.4, Riga LV-1048, Lavtia; e-mail: agris@cs.rtu.lv*

LOM MANAGER: УПРАВЛЕНИЕ ОБУЧАЮЩИМИ ОБЪЕКТАМИ В СИСТЕМЕ ПРОТОТИПИРОВАНИЯ ОБУЧАЮЩИХ КУРСОВ VITA II

Ольга Малиновская

Аннотация. *В статье описывается архитектура программного инструментария разработчиков адаптивных обучающих систем VITA II; адаптация в котором происходит на основе обучающих объектов, представленных в виде онтологий. Дается краткий обзор состава и выполняемых функций основных модулей системы. Подробно рассматривается структура модуля управления обучающими объектами LOM Manager.*

Ключевые слова*: E-learning, adaptive, ontologies, user modeling, Learning Object, LOM.*

ACM Classification Keywords*: K.3.1 Computer Uses in Education – Distance learning, K.3.2 Computer and Information Science Education – Computer science education, E.1 Data Structures – Trees, I.2.6 Learning – Concept learning (Knowledge acquisition).*

Введение

В последние годы в связи с широким развитием web-технологий в сети Internet доступно огромное количество научно-познавательной информации и тематических обучающих ресурсов.

Однако существует ряд проблем, касающихся размещения информационно-познавательных ресурсов в сети Internet. Основные проблемы состоят в том, что обилие информации, находящейся в сети, никак не связно и может много раз дублироваться. Это приводит к неэкономии как системных ресурсов (дискового объема сервера, трафика Internet), так и к нерациональному расходованию времени пользователя, осуществляющего поиск необходимой информации в сети Internet.

В последнее время Internet-сообщество все больше склоняется к идее многоразового использования обучающих ресурсов, в связи с этим была выдвинута концепция «обучающих объектов». Обучающий объект – это объект, содержащий как обучающую информацию по определенной тематике, так и служебную информацию в определенном стандартизированном формате. Обучающие объекты возможно повторно использовать и загружать в различные обучающие системы, которые по содержащейся в обучающемся объекте информации могут легко распознать его содержимое и подключить его в состав курса.

В данной работе рассматривается архитектура системы VITA II - универсального программного инструментария разработчика обучающих систем, позволяющего создать адаптивный обучающий курс, поддерживающий работу с обучающими объектами[1]. Рассматривается структура модуля системы VITA II, осуществляющего управление обучающими объектами, этот модуль был назван LOM Manager.

Архитектура системы VITA II

VITA II наследует основные преимущества инструментария «VITA» [1], в частности наличие представления данных в двух видах – в форме группы параграфов (модульное обучение) и онтологии (сетевое обучение).

Основным отличием системы VITA II от её предшественницы является поддержка многоуровневых онтологий.

Для поддержки многоуровневых вложенных онтологий была предложена следующая схема разделения онтологий на три уровня:

a. Фрагмент контента (Content fragment (CF) [2] – онтологии нижнего уровня (в листьях содержат конечные концепты, которые не могут быть раскрыты в онтологию – например, текст, видео, аудио, изображение, таблица);
b. Объект контента (Content Object (CO) [2] – онтологии среднего уровня (содержат набор CF, CO и навигацию);
c. Обучающий объект (Lerning object (LO) [2] – онтологии верхнего уровня (коллекции CO и связи между ними).

При этом онтологии верхнего уровня (Learning object) описываются по стандартам IEEE 1484.12.1 [3] – 2002 и ADL SCORM Version 1.3 [4], то есть должны быть легко расширяемы, доступны другими обучающими системами и независимы от программной платформы и контекста.

Проектируемый программный инструментарий предполагает возможность работы пользователя с разными типами информации, например текст, html страницы, изображения, таблицы, диаграммы, онтологии, мультимедиа и другие. При этом каждому типу ресурсов должен соответствовать свой программный модуль, осуществляющий работу с данным ресурсом.

Полный список требований к проектируемой системе VITA II описан в [5].

Для воплощения в проектируемом инструментарии заданных требований, была разработана следующая архитектура – в состав системы входят четыре основных модуля:

- модуль управления информационными ресурсами (Resource Manager);
- модуль управления обучающими объектами (LOM Manager);
- модуль управления онтологиями (Ontology Manager);
- модуль адаптации (Adaptation Manager) – привязка групп и онтологий к стереотипной модели пользователя.

[1] Данная работа ведется в рамках российско-белорусского проекта **ТАИС 06-01-81005**

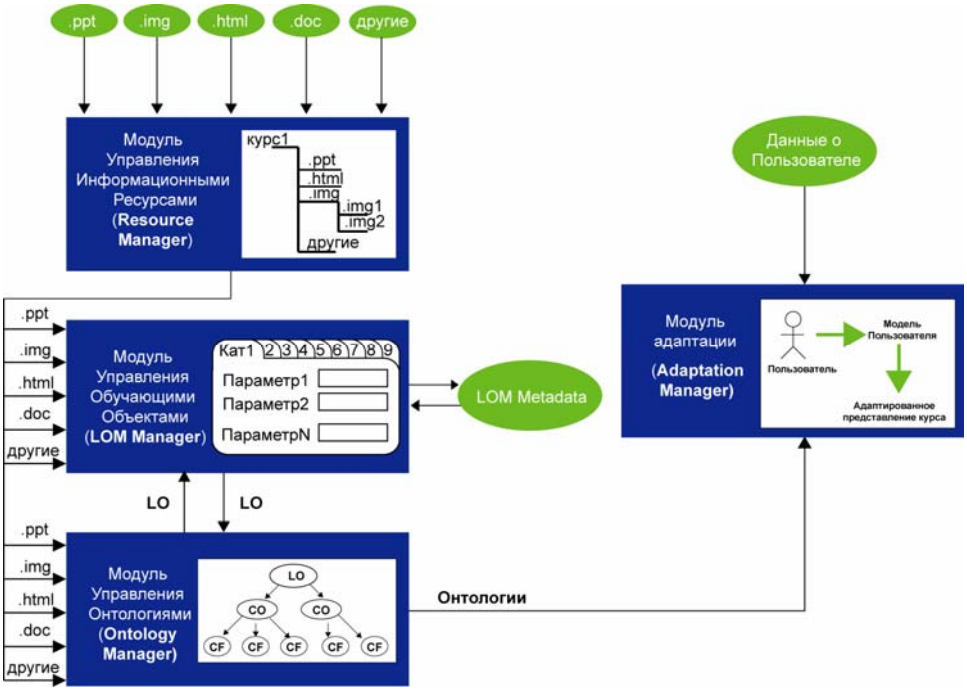


Рисунок 1. Архитектура системы VITA II

Полная архитектура системы представлена на Рисунке 1, краткое описание каждого модуля находится ниже.

Модуль управления информационными ресурсами (Resource Manager)

Модуль управления информационными ресурсами представляет из себя древовидную структуру, где элементами верхнего уровня детализации являются разделы типов ресурсов, например изображения, текст, гипертекст, аудио, видео, а элементами нижнего уровня – сами обучающие ресурсы.

В модуль управления информационных ресурсов происходит загрузка исходных данных обучающего курса, таких как текст, html-страница, изображения различных форматов, презентации, аудио, видео и другие ресурсы.

Помимо функции организации информационных ресурсов менеджер информационных ресурсов позволяет формировать группы («скелет» поля знаний обучающего курса). Формирование базиса курса осуществляется при помощи добавления узлов (параграфов) к структуре курса и определения их параметров. Добавляемый элемент является потомком узла, к которому он добавляется. Каждый узел впоследствии может быть раскрыт как глава обучающего курса (html-текст), несущая определённую справочную информацию.

Разработчик курса определяет требования для фрагмента страницы и для страницы в целом. Требования являются булевскими выражениями над значениями концептов. Когда требования для фрагмента страницы принимают значение истина, тогда фрагмент включается для отображения в курсе. В ином случае фрагмент исключается. Также разработчик определяет для каждой группы значения набора оформительских параметров. Оформительские интерфейсные параметры характеризуют отображение информации на экране и степень участия в этом пользователя [7].

Все эти параметры должны быть однозначно определяемы по модели пользователя и иметь возможность динамически изменяться.

Модуль управления онтологиями (Ontology Manager)

Модуль управления онтологиями предоставляет возможности создания и визуализации многоуровневых онтологий, содержащих три уровня вложенности, структура которых была описана выше.

Для поддержки сетевой технологии обучения необходимо к каждой модели пользователя привязать онтологию, по которой будет осуществляться навигация по обучающему курсу.

Модуль Адаптации (Adaptation Manager)

Для того, чтобы разрабатываемая система была адаптивной необходимо, чтобы она обеспечивала формирование и коррекцию модели пользователя.

Модель пользователя представляет собой вектор значений, характеризующих данного пользователя. Эти значения динамически меняются в зависимости от поведения обучаемого. Наиболее общая интерпретация этих значений заключается в том, что значение определяет уровень знаний пользователя в определённой области курса.

Каждый раз, когда пользователь посещает страницу, значение коэффициента, соответствующего номеру страницы увеличивается и соответственно увеличивается объём знаний пользователя в изучаемой области. Модель пользователя должна ссылаться на уникальную группу и уникальную онтологию.

Модуль управления обучающими объектами (LOM Manager)

Модуль управления обучающими объектами LOM Manager рассмотрим более подробно. LOM Manager отвечает за загрузку и выгрузку метаописания обучающего объекта в универсальном стандартизированном виде.

Модуль управления обучающими объектами может функционировать в двух режимах:

- в режиме генерации метаописания ресурса,
- в режиме загрузки и распознавания метаописания обучающего ресурса.

При функционировании в режиме генерации метаописания обучающего объекта на вход модуля могут поступать ресурсы совершенно различной структуры, такие как презентации, изображения, гипертекстовые документы, текстовые документы, онтологии, другие ресурсы.

На выходе модуль управления обучающими объектами LOM Manager должен выдавать метаописание полученного на входе ресурса в унифицированном виде.

При функционировании модуля в режиме распознавания метаописания обучающего объекта на вход модуля поступает метаописание ресурса в виде xml файла, на выходе выдаются значения метаданных, описанные в метаописании ресурса.

Для того, чтобы обучающие объекты, представляющие из себя небольшие контекстно-независимые элементы курса, могли быть легко заимствованы и загружены в другие обучающие системы к ним необходимо присоединить их метаописание. Метаописание задается в соответствии с международными стандартами LOM[3] и SCORM[4].

С помощью стандарта LOM может быть унифицировано содержание метаописания обучающего объекта. Для стандартизации самой структуры и вида представления этой информации используется стандарт SCORM, хранящий данные метаописания объекта, заданные в соответствии со стандартом LOM, в виде XML файла.

Архитектура работы модуля LOM Manager системы Vita II представлена на Рисунке 2.

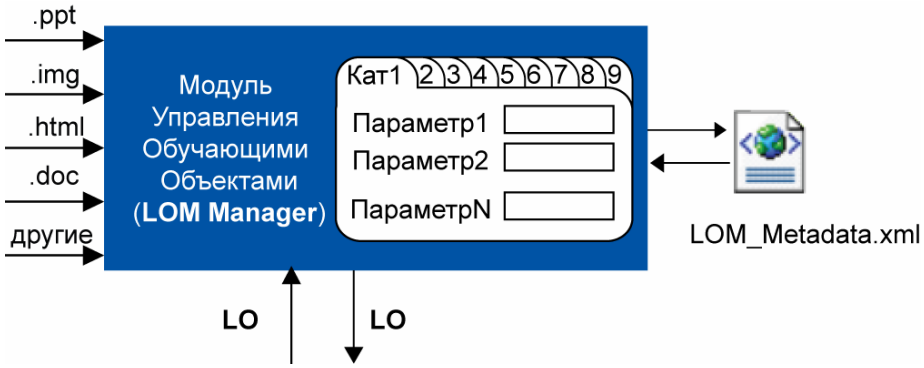


Рисунок 2. Архитектура модуля LOM Manager

Модуль управления обучающими объектами LOM Manager представляет собой диалоговое окно с девятью закладками. Каждая закладка содержит информацию об одной из девяти категорий, описанных в стандарте LOM [3].

На каждой закладке представлен список пар «Название поля формы»::Значение. Для формирования полного метаописания обучающего объекта в соответствии со стандартом LOM необходимо заполнить все поля, представленные на девяти закладках формы достоверной информацией, характеризующий описываемый обучаемый объект.

Далее рассматривается структура основных категория метаописания обучающего объекта.

Общая категория

Общая категория модели LOM группирует общую информацию об обучающем объекте, которая характеризует его в целом. Общая категория содержит следующие элементы данных[3]:

- Identifier – глобальный идентификатор, определяющий обучающий объект,
  - Catalog – название используемой для идентификации справочной или каталожной системы,
  - Entry – значение идентификатора в используемой системе каталогов, определяющее обучающий объект,
- Title - заголовок обучающего объекта,
- Language – человеческий язык (языки), используемый для общения с предположительным пользователем,
- Description – текстовое описание содержания обучающего объекта,
- Keyword – ключевое слово или фраза, описывающее тему обучающего объекта,
- Coverage – время, культура, география или регион, к которой относится данный обучающий объект,
- Structure – организационная структура обучающего объекта:
  - *Атом (Atomic)* – объект, неделимый в данном контексте,
  - *Коллекция (Collection)* – набор объектов без специальных связей между собой,
  - *Сеть (Networked)* – набор объектов взаимосвязанных неопределенными типами связи,
  - *Иерархия (hierarchical)* – набор объектов, взаимоотношения которых могут быть представлены деревом,
  - *Последовательность (linear)* – набор полностью упорядоченных объектов, то есть объединенных отношениями вида «следующий» и «предыдущий»
- Aggregation Level – функциональная детализация данного обучающего объекта:
  - 1: наименьший уровень агрегации: *«сырые» медиа-данные или фрагменты*
  - 2: коллекция обучающих объектов 1 уровня: *урок*
  - 3: коллекция обучающих объектов 2 уровня: *курс*
  - 4: наивысший уровень детализации: *набор курсов для сдачи сертификационного экзамена.*

Категория Жизненного цикла

Категория Жизненного цикла модели LOM [3] содержит факты, связанные с историей и текущим состоянием обучающего объекта, а также информацию о том, что повлияло на описываемый обучающий объект в процессе его эволюции. Данная категория содержит [3]:

- Version – версия обучающего объекта,
- Status – статус обучающего объекта,
- Contribute – набор факторов (люди, организации и т.д.), повлиявших на существо обучающего объекта в течение его жизненного цикла (создание, редактирование, публикации и т.д.). Внутри данного элемента указываются следующие параметры:
  - Role – роль:
    - Автор;
    - Издатель;
    - Неизвестный;
    - Инициатор;
    - Редактор;
    - Графический дизайнер.
  - entity – идентификация и информация о факторах (людях, организациях и т.д.), внесших вклад в развитие обучающего объекта. Факторы должны быть упорядочены в порядке убывания их значимости,
  - Date – дата влияния на обучающий объект.

Категория Meta-Metadata

Категория Meta-Metadata модели LOM [3] содержит информацию о самих метаданных, в неё входят следующие элементы:

- Identifier - уникальный глобальный идентификатор, определяющий запись о структуре метаданных,
  - Catalog - имя системы классификации или идентификации, используемой для определения обучающего объекта,
  - Entry – запись, описывающая метаданные обучающего объекта,
- Contribute – набор факторов (люди, организации и т.д.), повлиявших на существо метаданных обучающего объекта в течение его жизненного цикла (создание, редактирование, публикации и т.д.). Внутри данного элемента указываются следующие параметры:
  - Role – роль:
    - создатель
    - корректор
  - entity – идентификация и информация о факторах (людях, организациях и т.д.), внесших вклад в развитие метаданных обучающего объекта. Факторы должны быть упорядочены в порядке убывания их значимости,
  - Date – дата влияния на метаданные обучающего объекта.

Техническая категория

Техническая категория содержит технические требования и характеристики, необходимые для работы с обучающим объектом. В эту категорию входят следующие параметры:

- Format - Этот элемент данных используется для определения технических требований, необходимых для работы с обучающим объектом,
- Size - Размер обучающего объекта в байтах (указывается актуальный размер несжатого объекта),
- Location - Строка, указывающая путь к обучающему объекту. Это может быть место размещения (Universal Resource Locator) или метод (Universal Resource Identifier),
- Requirement - Технические требования, необходимые для работы с обучающим объектом,
- Installation Remarks - Описание процесса установки обучающего объекта.

Образовательная категория

Данная категория описывает образовательные и педагогические характеристики обучающего объекта, содержит следующие элементы:

- Interactivity Type - Преобладающий тип обучения, используемый в обучающем объекте:
  - Активное (active)
  - Описательное (expositive)
  - Смешанное (mixed)
- Learning Resource Type - Определяет специфический тип обучающего объекта,
- Interactivity Level - Уровень интерактивности, характеризующий данный объект,
- Semantic Density - Степень выразительности обучающего объекта,
- Intended End User Role - Модель (модели) пользователя, на которую в первую очередь рассчитан обучающий объект,
- Contex - Окружающая среда, в которой предполагается использование обучающего объекта,
- Typical Age Range - Возраст предполагаемого пользователя обучающего объекта,
- Difficulty - Степень сложности работы с обучающим объектом,
- Typical Learning Time - Предположительное время, необходимое для изучения обучающего объекта,
- Description - Комментарии о том, как обучающий объект должен быть использован,
- Language – язык,

Категория Прав Доступа

Категория Прав Доступа содержит информацию об интеллектуальных правах собственности использования обучающего объекта, в неё входят:

- Cost – определяет, необходимо ли платить за использование обучающего объекта (да/нет),
- Copyright and Other Restrictions – защищен ли обучающий объект авторскими правами (да/нет),
- Description – комментарии, описывающие правила использования обучающего объекта.

Категория Отношений

Категория Отношений определяет отношения между LO и другие связные LO, в неё входят:

- Kind – тип отношений между данным обучающим объектом и целевым объектом, описанном в виде Relation.Resource,
- Resource – целевой обучающий объект, на который ссылается данный обучающий объект,
  - Identifier – уникальный идентификатор, определяющий целевой обучающий объект,
  - Description – описание целевого обучающего объекта

Категория Аннотаций

Категория Аннотаций содержит комментарии об использовании обучающего объекта в обучении и сохранит информацию о том, кто и когда создавал эти комментарии. В эту категорию входят:

- Entity – сообщество (люди, организация и т.д.), которые создали данную аннотацию,
- Date – дата создания аннотации,
- Description – содержание аннотации.

Категория Классификации

Категория Классификации описывает обучающий объект и его местоположение в классификационной структуре, содержит:

- Purpose – цель классификации обучающего объекта,
- Taxon Path – таксономический путь в специализированной системе классификации,
  - Source – имя системы классификации,
  - Taxon – отдельная часть (терм) таксономии,
  - Description – описание, соответствующее целям классификации,
  - Keyword – ключевые слова и фразы, соответствующие целям классификации.

Выводы

В результате проделанной работы была разработана система прототипирования адаптивных обучающих курсов, поддерживающая использование обучающих объектов. Информация, содержащейся в курсе, реализованном с помощью описанной системы, может быть легко импортирована и экспортирована в другие обучающие информационные системы.

Обучающие материалы соответствуют международным стандартам, являются легко расширяемыми, платформо-независимыми, пополняемыми, могут повторно использоваться в других системах. Тем самым реализована идея стандартизации и унификации обучающих ресурсов в сети Internet

Литература

1. Гаврилова Т.А, Гелеверя Т.Е. «Программный инструментарий Vita. Версия 2.1. Техническая документация», 2002

2. Jovanovic' J., Gasevic' D., "Ontology of Learning Object Content Structure", 2005

3. Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 15 July 2002

4. ADL SCORM Version 1.3 WORKING DRAFT 0.9, November 27, 2002

5. Гелеверя Т., Малиновская О., Гаврилова Т., Курочкин М., «Система VITA-II для прототипирования учебных курсов on-line на основе онтологий», сборник конференции «MEL-2006»

6. Брусиловский П. Л., 1996 "Технологии и методы адаптивной гипермедиа", User Modeling and User Adapted Interaction, v 6, n 2-3, стр. 87-129, http://ifets.ieee.org/russian/depository/Brusil_1996.zip

7.Васильева Е. "Проблемы интерфейсной адаптации в обучающих системах"

Authors' Information

Olga Malinovskaya *– post-graduate student of Saint-Petersburg State Polytechnical University, Politechnicheskaya 29, 195251, St. Petersburg, Russia, e-mail: malinol@mail.ru*

THE NEW SOFTWARE PACKAGE FOR DYNAMIC HIERARCHICAL CLUSTERING FOR CIRCLES TYPES OF SHAPES

Tetyana Shatovska, Tetiana Safonova, Iurii Tarasov

Abstract*: In data mining, efforts have focused on finding methods for efficient and effective cluster analysis in large databases. Active themes of research focus on the scalability of clustering methods, the effectiveness of methods for clustering complex shapes and types of data, high-dimensional clustering techniques, and methods for clustering mixed numerical and categorical data in large databases. One of the most accuracy approach based on dynamic modeling of cluster similarity is called Chameleon. In this paper we present a modified hierarchical clustering algorithm that used the main idea of Chameleon and the effectiveness of suggested approach will be demonstrated by the experimental results.*

Keywords*: Chameleon, clustering, hypergraph partitioning, coarsening hypergraph.*

ACM Classification Keywords *F.2.1 Numerical Algorithms and Problems*

Introduction

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to

the objects in other clusters. A cluster of data objects can be treated collectively as one group in many applications. Data clustering is under vigorous development. Contributing areas of research include data mining, statistics, machine learning, spatial database technology, biology, and marketing. Owing to the huge amounts of data collected in databases, cluster analysis has recently become a highly active topic in data mining research. As a branch of statistics, cluster analysis has been studied extensively for many years, focusing mainly on distance-based cluster analysis. Active themes of research focus on the scalability of clustering methods, the effectiveness of methods for clustering complex shapes and types of data. Chameleon is a clustering algorithm that explores dynamic modeling in hierarchical clustering. In its clustering process, two clusters are merged if the interconnectivity and closeness between two clusters are highly related to the internal interconnectivity and closeness of objects within the clusters. The merge process based on the dynamic model facilitates the discovery of natural and homogeneous clusters and applies to all types of data as long as a similarity function is specified. Chameleon is derived based on the observation of the weakness of two hierarchical clustering algorithms: CURE and ROCK. CURE and related schemes ignore information about the aggregate interconnectivity of objects in two different clusters, whereas ROCK and related schemes ignore information about the closeness of two clusters while emphasizing their interconnectivity. In this paper, we present our experiments with hierarchical clustering algorithm CHAMELEON for circles cluster shapes with different densities using hMETIS program that used multilevel k-way partitioning for hypergraphs and a Clustering Toolkit package that merges clusters based on a dynamic model. In CHAMELEON two clusters are merged only if the inter-connectivity and closeness between two clusters are comparable to the internal inter-connectivity of the clusters and closeness of items within the clusters. The methodology of dynamic modeling of clusters is applicable to all types of data as long as a similarity matrix can be constructed. We present a modified hierarchical clustering algorithm that measures the similarity of two clusters based on a new dynamic model with different shapes and densities. The merging process using the dynamic model presented in this paper facilitates discovery of natural and homogeneous not only circles cluster shapes.

1 Related work

In this section, we give a brief description of existing clustering algorithms.

A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed. The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups close to one another, until all of the groups are merged into one, or until a termination condition holds. The divisive approach, also called the top-down approach, starts with all the objects in the same cluster. In each successive iteration, a cluster is spitted up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.

Hierarchical methods suffer from the fact that once a step is done, it can never be undone. This rigidity is useful in that it leads to smaller computation costs by not worrying about a combinatorial number of different choices. However, a major problem of such techniques is that they cannot correct erroneous decisions. There are two approaches to improving the quality of hierarchical clustering: (1) perform careful analysis of object "linkages" at each hierarchical partitioning, such as in CURE and Chameleon, or (2) integrate hierarchical agglomeration and iterative relocation by first using a hierarchical agglomerative algorithm and then refining the result using iterative relocation, as in BIRCH [Zhang, 1996].

Most clustering algorithms either favor clusters with spherical shape and similar sizes, or are fragile in the presence of outliers. CURE overcomes the problem of favoring clusters with spherical shape and similar sizes and is more robust with respect to outliers. CURE employs a novel hierarchical clustering algorithm that adopts a middle ground between centroid-based and representative-object-based approaches. Instead of using a single centroid or object to represent a cluster, a fixed number of representative points in space are chosen. The representative points of a cluster are generated by first selecting well-scattered objects for the cluster and then "shrinking" or moving them toward the cluster center by a specified fraction, or shrinking factor. At each step of

the algorithm, the two clusters with the closest pair of representative points (where each point in the pair is from a different cluster) are merged. ROCK is an alternative agglomerative hierarchical clustering algorithm that is suited for clustering categorical attributes. It measures the similarity of two clusters by comparing the aggregate interconnectivity of two clusters against a user-specified static interconnectivity model, where the interconnectivity of two clusters is defined by the number of cross links between the two clusters, and link is the number of common neighbors between two points. In other words, cluster similarity is based on the number of points from different clusters who have neighbors in common [Guha, 1999].

ROCK first constructs a sparse graph from a given data similarity matrix using a similarity threshold and the concept of shared neighbors. It then performs a hierarchical clustering algorithm on the sparse graph.

There are two major limitations of the agglomerative mechanisms used in existing schemes. First, these schemes do not make use of information about the nature of individual clusters being merged. Second, one set of schemes (CURE and related schemes) ignore the information about the aggregate interconnectivity of items in two clusters, whereas the other set of schemes ignore information about the closeness of two clusters as defined by the similarity of the closest items across two clusters.

2 Overview of CHAMELEON: Clustering Using Dynamic Modeling

Chameleon is a clustering algorithm that explores dynamic modeling in hierarchical clustering [Karypis, 1999a]. Chameleon represents its objects based on the commonly used k-nearest neighbor graph approach. This graph representation of the data set allows CHAMELEON to scale to large data sets. Each vertex of the k-nearest neighbor graph represents a data object, and there exists an edge between two objects if one object is among the k-most similar objects of the other. The k-nearest neighbor graph captures the concept that neighborhood radius of an object is determined by the density of the region in which this object resides [Mitchell, 1997].

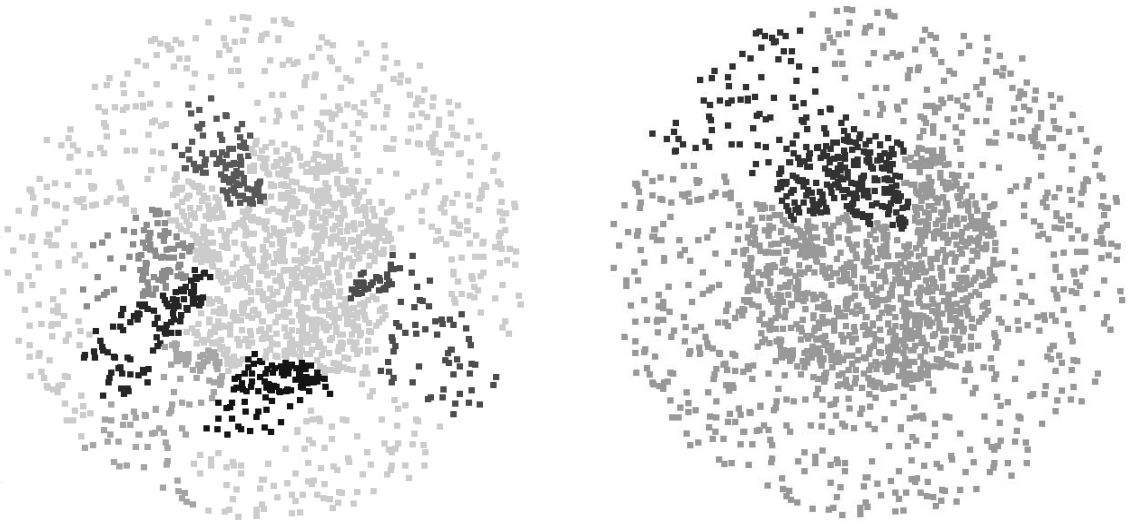
During the next step a sequence of successively smaller hypergraphs are constructed – Coarsening Phase. Two primary schemes have been developed for selecting what groups of vertices will be merged together to form single vertices in the next level coarse hypergraphs. The first scheme called edge-coarsening (EC) [Alpert, 1997] selects the groups by funding a maximal set of pairs of vertices (i.e., matching) that belong in many hyperedges. The second scheme that is called hyperedge-coarsening (HEC) [Karypis, 1997] finds a maximal independent set of hyperedges, and the sets of vertices that belong to each hyperedge becomes a group of vertices to be merged together. At each coarsening level, the coarsening scheme stop as soon as the size of the resulting coarse graph has been reduced by a factor of 1.7 [Karypis, 1999b]. The third phase of the algorithm is to compute a k-way partitioning of the coarsest hypergraph such that the balancing constraint is satisfied and the partitioning function as mincut is optimized. During the uncoarsening phase, a partitioning of the coarser hypergraph is projected to the next level finer hypergraph, and a partitioning refinement algorithm is used to optimize the objective function without violating the partitioning balancing constraints. At the final iteration of algorithm CHAMELEON determines the similarity between each pair of clusters by taking into account both at their relative inter-connectivity and their relative closeness. It selects to merge clusters that are well inter-connected as well as close together with respect to the internal inter-connectivity and closeness of the clusters. By selecting clusters based on both of these criteria, CHAMELEON overcomes the limitations of existing algorithms that look either at the absolute inter-connectivity or absolute closeness.

3 Performance Analysis

The overall computational complexity of CHAMELEON depends on the amount of time it requires to construct the K – nearest neighbors graph and the amount of time it requires to perform the two phases of the clustering algorithm. In [Karypis, 1999a] was shown that CHAMELEON is not very sensitive of values k for computing the k-nearest neighbor graph, of the value of MINSIZE for the phase I of the algorithm, and of scheme for combining relative inter-connectivity and relative closeness and associated parameters, and it was able to discover the correct clusters for all of these combinations of values for k and MINSIZE. In this section, we present

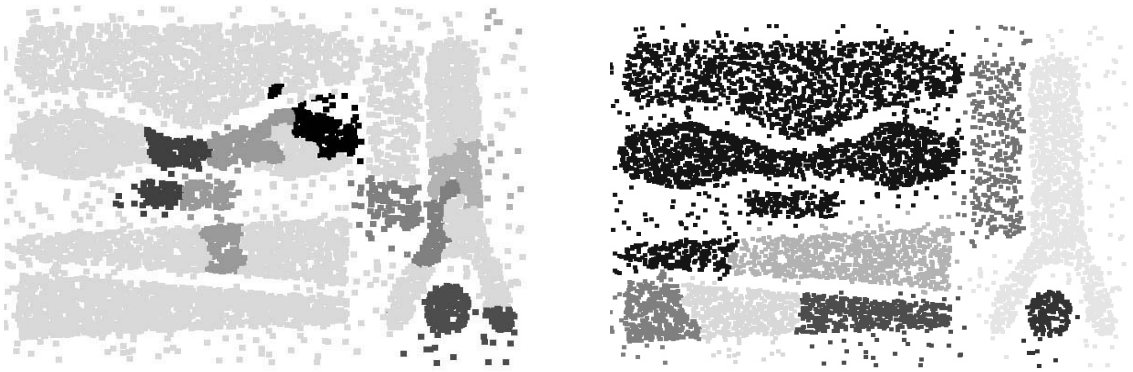
experimental evaluation of clustering using hMETIS hypergraph partitioning package for k-way partitioning of hypergraph and for recursive bisection [Karypis, 1998] and CLUTO 2.1.1– A Clustering Toolkit [Karypis, 2003].

We experimented with five different data sets containing points in two dimensions: “disk in disk”, t4.8k, t5.8k, t8.8k, t7.10k [Karypis lab.].The first data set, has a particularly challenging feature that two clusters are very close to each other and they have different densities and circles shapes. We choose the number of neighbors k=5, 15, 40, MINSIZE = 5%. Looking at Fig. 1, a) we can see the results of the k-way partitioning of hypergraph by hMETIS package [Karypis lab.] and b) merging process by CLUTO package [Karypis lab.] with k=5 nearest neighbors. Looking at Fig.1 we can see that in both cases we have not correctly identified the genuine clusters.



a) k-way partitioning by hMETIS b) final clusters by CLUTO

Fig. 1 Data set “disk in disk” with k=5 nearest neighbors and asymmetric k-NN



a) k-way partitioning by hMETIS b) final clusters by CLUTO

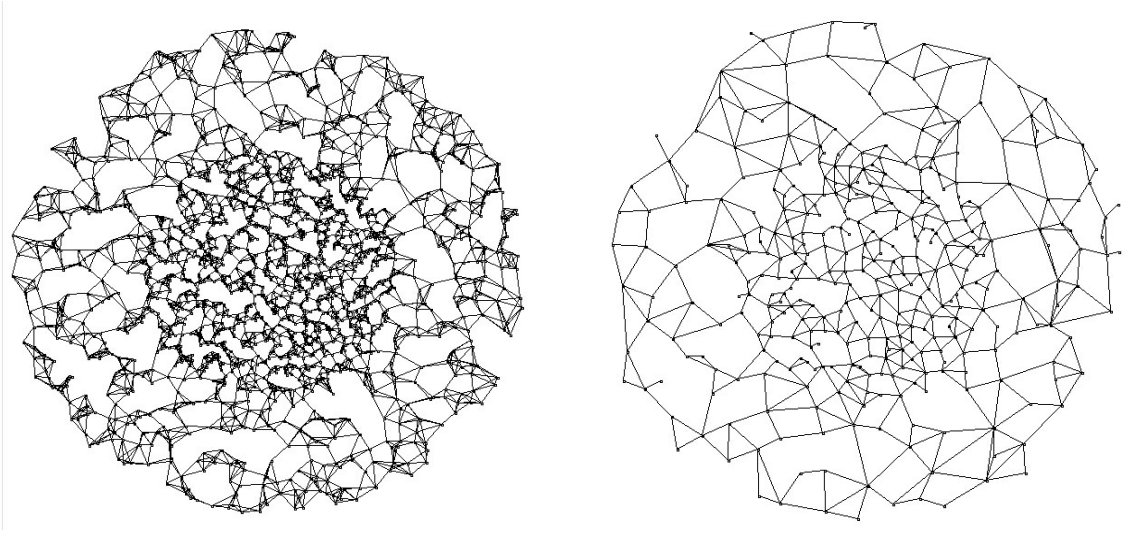
Fig. 2 Data set “t8.8k” with k=5 nearest neighbors and asymmetric k-NN

The data set t8.8k has eight clusters of different shapes, size and orientation, some of which are inside the space enclosed by other clusters. Moreover, it also contains random noise such as a collection of points forming vertical streaks. Looking at Fig. 2 with k=5 nearest neighbors we can see that hMETIS also compute k-way partitioning of hypergraph with mistakes closer to the border of two classes and CLUTO can not effectively merge clusters for such type of dataset using asymmetric k-NN, with k=5. It means that algorithm of the partitioning phase is very sensitive to the value of k for spherical shapes of clusters and to the types of k-NN graph (symmetric and asymmetric). It is very important to choose an optimal value of k, because with k=16 and more, and only for symmetric k-NN with weights of edges equal to the number of common neighbors we obtain final clustering with minimum percentages of errors.

4 Modeling the cluster similarity

As we remark above the CHAMELEON operates on a sparse graph in which nodes represent data items and weighted edges represent similarities among the data item (symmetric graph) [Karypis, 1999a]. In our algorithm during first phase we construct an asymmetric k-NN graph and there exists an edge between two points if for one of it there exist closest neighbor among all existing neighbors according to the value of k. Note that the weight of an edge connecting two objects in the k-NN graph is a similarity measure between them, as usual a simple distance measure (or inversely related to their distance).

In our algorithm the weight of an edge we compute as weighted distance between objects. Fig. 3 represents the k-NN graph for data set “disk in disk” with k=5. During coarsening phase the set of smaller hypergraphs is constructed. In the first stage of coarsening process we choose the set of vertices with maximum degrees and matched it with a random neighbour. On the other stages we visit each vertex in a random order and matched it with adjacent vertex via heaviest edge. Note that usually the weight of an edge connecting two nodes in a coarsened version of the graph is the number of edges in the original graph that connect the two sets of original nodes collapsed into the two coarse nodes. In our case we compute the weight of the hyperedge as the sum of the weights of all edges that collapse on each other during coarsening step. We stop the coarsening process at each level as soon as the number of multivertices of the resulting coarse hypergraph has been reduced by a constant less than 2 (Fig. 3).



a) k-NN graph with k=5 b) hypergraph after third level of coarsening

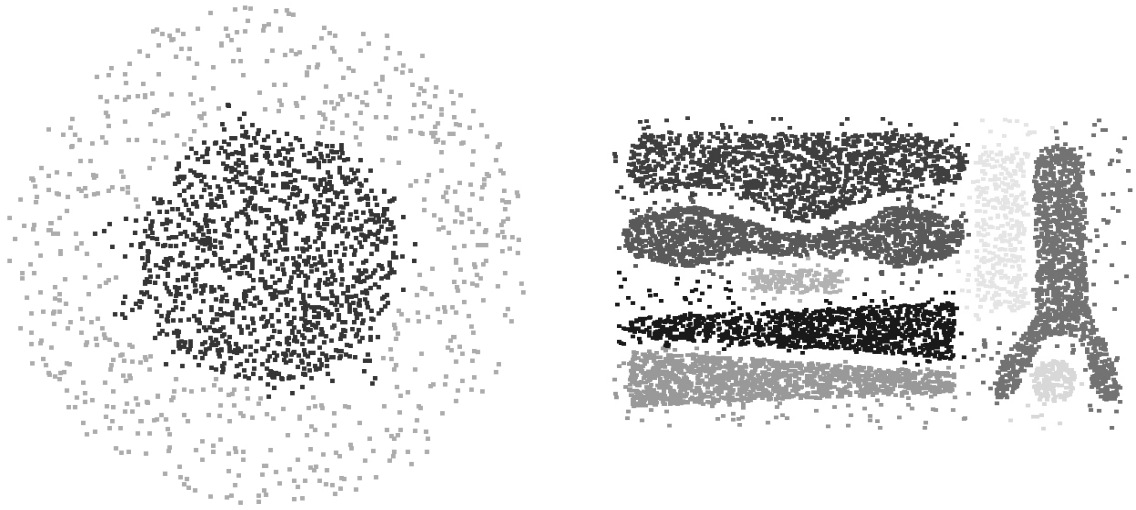
Fig. 3 Data set “disk in disk”

On the next level of algorithm we produce a set of small hypergraphs using k-way multilevel paradigm [Karypis, 1999b]. We start the process of partitioning by choosing k most heavier multivertices, where k = 8, 16, 32. After that we gathering one by one all neighbors from each chosen vertex and obtain the initial partitioning w.r.t. to the balancing constant. The problem of computing an optimal bisection of a hypergraph is NP-hard. One of the most commonly used objective function is to minimize the hyperedge-cut of the partitioning; i.e., the total number of hyperedges that span multiple partitions [Karypis, 1999b]. One of the most accuracy algorithm of partitioning the hypergraph is Kernighan-Lin / Fiduccia – Mattheyses algorithm, in which during each pass, the algorithm repeatedly finds a pair of vertices, one from each of the subdomains, and swaps their subdomains. The pairs are selected so as to give the maximum improvement in the quality of the partitioning even if this improvement is negative. Once a pair of vertices has been moved, neither are considered for movement in the rest of the pass. When all of the vertices have been moved, the pass ends. At this point, the state of the bisection at which the minimum edge-cut was achieved is restored. In our experiments we use a greedy refinement algorithm developed by George Karypis [Karypis, 1999b], but as the gain function for each vertex we compute the differences between

the sum of the weights of edges incident on vertex that go to the other partition and the sum of edges weights that stay within the partition. We choose the vertex with maximum positive gain and move it if it result in a positive gain, so we work only with boundary vertices.

After the partitioning of hypergraph into the large number of small parts we start to merge the pair of clusters for which both relative inter-connectivity and their relative closeness are high [Karypis, 1999a]. In our research we use George Karypis formula to compute the similarity between sub-clusters. Looking at the Fig.1 b) we can see that for data set "disk in disk" was obtained not correct clustering results. Thus we suggest to modified the above mentioned expression by change the relative inter-connectivity to a new expression that estimates the average weights of edges in each sub-graph and the number of edges that connect two partitions to the number of edges that stay within the smallest partition.

Looking at the Fig. 4 we can see the correct clustering results for the same data set "disk in disk" using our suggested expression. For another above mentioned data sets we obtain as well accuracy results. In all experiment we use k=5 and in our approach the correctness of classification really doesn't depend of the value of k and of the k-NN type.



a) Data set "disk in disk" b) Data set "t8.8k.txt"

Fig 4. Clustering results using a new approach to the sub-clusters merging, k=5

Conclusion

In this paper, we present our experiments with hierarchical clustering algorithm CHAMELEON for circles cluster shapes with different densities using hMETIS program that used multilevel k-way partitioning for hypergraphs and the Clustering Toolkit package that merges clusters based on a dynamic model. In CHAMELEON two clusters are merged only if the inter-connectivity and closeness between two clusters are comparable to the internal inter-connectivity of the clusters and closeness of items within the clusters. The methodology of dynamic modeling of clusters is applicable to all types of data as long as a similarity matrix can be constructed.

Experimental results showed that hMETIS computes k-way partitioning of hypergraph with mistakes closer to the border of two classes and CLUTO can not effectively merge clusters using asymmetric k-NN, with k=5.

We present a modified hierarchical clustering algorithm that measures the similarity of two clusters based on a new dynamic model with different shapes and densities. The merging process using the dynamic model presented in this paper facilitates discovery of natural and homogeneous not only circles cluster shapes.

Experimental results showed that this method is not sensitive to the value of k and doesn't need a specific k-nearest neighbor graph creating.

Bibliography

[Alpert, 1997] C. J. Alpert, J. H. Huang and A. B. Kahng, Multilevel circuit partitioning. In: Proc. of the 34th ACM/IEEE Design Automation Conference. 1997.

[Guha, 1999] S. Guha, R. Rastogi, K. Shim ROCK: Robust Clustering using linKs, (ICDE'99).

[Karypis, 1997] G. Karypis , R. Aggarwal, V. Kumar and Sh. Shekhar, Multilevel hypergraph partitioning: Application in VLSI domain. In: Proceedings of the Design and Automation Conference. 1997.

[Karypis, 1998] G. Karypis, and V. Kumar , hMETIS 1.5.3: A hypergraph partitioning package. Technical report. Department of Computer Science, University of Minnesota, 1998

[Karypis, 1999a] G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithms Using Dynamic Modeling. IEEE Computer, 32(8):68–75, 1999.

[Karypis, 1999b] G. Karypis and V. Kumar. Multilevel k-way hypergraph partitioning. In Proceedings of the Design and Automation Conference, 1999.

[Karypis, 2003] G. Karypis, CLUTO 2.1.1. A Clustering Toolkit. Technical report. Department of Computer Science, University of Minnesota, 2003

[Karypis lab.] http://www.cs.umn.edu/~karypis.

[Mitchell, 1997] T. M. Mitchell. Machine Learning. McGraw Hill, 1997

[Zhang, 1996] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.

Authors' Information

Shatovska Tetyana – *Kharkiv National University of Radio Electronics, Computer Science department, P.O.Box: Kharkiv-116, Lenin av. 14, Ukraine; e-mail: tanita_uk@mail.ru*

Safonova Tetyina - *Kharkiv National University of Radio Electronics, Computer Science department, P.O.Box: Kharkiv-116, Lenin av. 14, Ukraine; e-mail: safonovatm@mail.ru*

Tarasov Iuril *- Kharkiv National University of Radio Electronics, Computer Science department, P.O.Box: Kharkiv-116, Lenin av. 14, Ukraine; e-mail: ytarasovmail@rambler.ru*

THE METRICS AND MEASURE OF REFUTABILITY ON FORMULAS IN THE THEORY T

Alexandr Vikent'ev

Abstract: *The paper discusses statements of experts about objects represented as the formulas in language L=L(**T**) some theory **T** and offers techniques for introducing metrics on such statements and measure of refutability. The research will find a use to problems of the best matching of the statements, of construction the decision functions of pattern recognition and development of expert's systems. The offered refutability functions satisfy all requirements (for informativeness) formulated in [1, 2].*

Keywords*: pattern recognition, distance between statements, theory of models, metrics, know base.*

ACM Classification Keywords*: I.2.6. Artificial Intelligence - knowledge acquisition.*

Introduction

The theory and methods of constructing decision function of pattern recognition on the basis of an analysis of empirical information represented as tabulated data have been well advanced by now. In addition to this there is

an increasing interest in the construction of decision function on the basis of an analysis of experts information provided in the form of logical "knowledge" of several experts which are represented by the predicate formulas [1-2, 5-6] (these "knowledge" can be partly or completely contradictory) in some L=L(T). It involves the problem of matching the statements of the experts about hierarchical objects and also the problems of introducing a distance on these statements and of defining their refutability. Obviously the statements ("knowledge") can differ on quantity of refutability contained in them. The refutability reflects importance of the information informed by the expert.

This work is a natural continuation of works [1-2] and the familiarity with them is supposed. The paper discusses logical statements of experts about hierarchical objects recorded as the logical predicate formulas. By making use of the methods of mathematical logic and of the model theory we offer the techniques for introducing metrics on these statements and measure of their refutability . We study the properties of entered metrics and connected to them measures. The work was supported by the Russia Foundation for Basic Research № 07-01-00331 and by program "Logika" of Novosibirsk State University.

The distances on the formulas and its properties

Let L = L(*T*) be a first order language consisting of final number of predicate symbol which are selected for record and study the connections between variables in particular application area. For each variable x_i there is the unary predicate P_{x_i} determining only on the range of values of variable x_i on the set A_n. Let A_n be nonempty set of power n ($\leq n$). A_n is the join of all values of considering variables which are included in the predicates. In many applied problems the value of variables and their number are final and so the model with final set are considered.

Definition 0.[5-6]. The interpretation γ is the map putting in conformity to each symbol $P_i^{n_i}$ of signature Ω of language L ($\Omega = \left\langle P_1^{n_1}, P_2^{n_2}, \ldots \right\rangle$, where $P_i^{n_i}$ - n_i-ary predicate) particular n_i-ary relation $P_i^{A_n}$ determined on the set A_n. It allows to speak about model $\left\langle A_n; \Omega^{A_n} \right\rangle$ of signature Ω. Let's consider models only of final signature.

Let there is a final number s of the experts and the area of possible values of variables. The models (in sense of model theory) are set by experts. Each expert j sets the interpretation of each predicate symbol $P_i^{n_i}$ of the language L by the appropriate relation $\gamma\left(P_i^{n_i}\right) = P_i^{\mathsf{M}_j} \subseteq A_n^{n_i}$ in model M_j. Then we have "knowledge" of the experts recorded as the formula which are set by formula's subset in each model M_j [4-5] under the interpretation of the expert j.

Let $Mod_n(L)$ be a set of all models T of the language L (T) determined on the set A_n by the experts.

We shall consider σ - algebra of subset F on the set $\mathrm{A} = A_n^{<\omega} = \bigcup_{k<\omega} A_n^k$ ($A_n^k = A_n \times \ldots \times A_n$). Only those subset S_j from F will be interested us for which the formula ψ of language L, appropriated to "knowledge" of the experts, will be discovered such that under the interpretation of "knowledge" of expert j ($j = 1,\ldots,s$) $S_j = \psi(\mathsf{M}_j) = \{\bar{a} \mid \mathsf{M}_j \models \psi(\bar{a})\}$ (that is formula's subset which is appropriated to the formula ψ in model M_j [5-6]).

Let U be a set of all predicate symbols used by the experts; B - is the closure of the set U under logical operations $\neg, \wedge, \vee, \rightarrow$ and quantors $\forall$ and $\exists$ on variables. Obviously, the set of formulas interesting for us is contained in B.

Definition 1. [7]. By the probabilistic measure μ on the set B we mean a function $\mu : B \to$ [0,1] satisfying conditions for ϕ and $\psi \in B$:

1) if $\models \phi \equiv \psi$, then $\mu(\phi) = \mu(\psi)$;

2) if $\models \phi$, then $\mu(\phi) = 1$;

3) if $\models \neg\phi$, then $\mu(\phi) = 0$;

4) $\mu(\neg\phi) = 1 - \mu(\phi)$;

5) if $\models \neg(\phi \wedge \psi)$, then $\mu(\phi \vee \psi) = \mu(\phi) + \mu(\psi)$.

Let a probabilistic measure μ is set on sets from F.

Instead of "knowledge" of the expert recorded as predicate $P^{\mathbf{M}_i}$ in model $\mathbf{M}_i$ further we shall consider it approximation – predicate $\widetilde{P}^{\mathbf{M}_i}$.

Under approximation $\widetilde{P}^{\mathbf{M}_i}$ of predicate $P^{\mathbf{M}_i}$ is understood closer definition of the domain of truth of this predicate in model $\mathbf{M}_i$ by one of ways:

1) to leave the relation without changes;
2) to eliminate these elements from $P^{\mathbf{M}_i}$ in which truth the expert i is not absolutely sure ;
3) to add in relation new elements and eliminate some old, for example, with allowance for the "knowledge" of other experts;
4) to execute items 2) and 3) simultaneously.

Let's enter a distance on a set of "knowledge" of the experts with the help of models which are set be them. The models differ by the interpretations.

Let's define a distance between the formula's subset (predicates) in each model for T $\mathsf{M}_i \in Mod_n(L)$ as a measure of their symmetrical difference.

Definition 2. We call $\rho_{\mathbf{M}_i}\left(P_k^{\mathbf{M}_i}, P_j^{\mathbf{M}_i}\right) = \mu\left(\widetilde{P}_k^{\mathbf{M}_i} \Delta \widetilde{P}_j^{\mathbf{M}_i}\right)$ the distance between predicates $P_k^{\mathbf{M}_i}$ and $P_j^{\mathbf{M}_i}$ determined in model $\mathbf{M}_i$.

Remark. This definition is correct if the predicates of equal arity and with an identical set of variables. If the considered predicates have different arity or different set of variables and if the expert consider insignificant the absent in one of the formulas variable x_i wt suppose that it receives anyone from possible values. Otherwise (if it is significant) we determine this variable by adding to the necessary formula the predicate P_{x_i} selecting values of this variable. It's clear that the entered concept is easily spreaded on formula's subset. Further we shall study a distance between the formulas of the same variables (equal arity).

The distance between the formulas determined on the set of models $Mod_n(L)$ we shall define as mean on a set of distances in models

Definition 3. We call $\rho_1\left(P_k(\bar{x}), P_j(\bar{x})\right) = \dfrac{\sum_{\mathsf{M}_i \in Mod_n(L)} \rho_{\mathsf{M}_i}\left(P_k^{\mathsf{M}_i}, P_j^{\mathsf{M}_i}\right)}{\left|Mod_n(L)\right|}$ the distance between the formulas $P_k(\bar{x})$ and $P_j(\bar{x})$ determined on the set $Mod_n(L)$.

Now we shall consider a way of definition of a distance between the sentences (closed formulas).

We denote by $Mod(\phi)$ the set of models in $Mod_n(L)$ on which the sentence ϕ is true, that is $Mod(\phi) = \{\mathbf{M}_i \in Mod_n(L) \mid \mathbf{M}_i \models \phi\}$.

Clear, there are such models for which the sentence is true and such models for which it is false (if it is not a tautology). It is natural to measure the difference of information contained in the sentences by the number of the models on which sentences have different truth values.

Definition 4. We call $\rho_2(\phi,\psi)=\frac{\left|Mod\left((\phi\wedge\neg\psi)\vee(\neg\phi\wedge\psi)\right)\right|}{\left|Mod_n(L)\right|}$ the distance between the sentences ϕ and ψ.

Let's consider one more way of definition of a distance between the formulas. Let's add the first order language L by constant from the set $M = Mod_n(L)$. For this set M we shall consider any tuples $\bar{a}$ which lengths are equal to arity of the formulas $l(\bar{a})$. At the substitution of tuples in the formulas in the supposition that the formulas have identical arity (as it to achieve was indicated above) the formulas become the sentences.

Definition 5. We call $\rho_3(\phi(\bar{x}),\psi(\bar{x}))=\min\limits_{\bar{a}\in M^{l(\bar{a})}}\rho_2(\phi(\bar{a}),\psi(\bar{a}))$ the distance between the formulas $\phi(\bar{x})$ and $\psi(\bar{x})$.

The following theorem is proved and from theorem follows that the offered distances are the metrics really. Some additional properties of entered distances are proved in the theorem.

Theorem 1. For any formulas ("knowledge" of the experts) ϕ, ψ, χ and for any function ρ_i on T the following assertions are valid:

1. $0\le\rho_i(\phi,\psi)\le 1$.
2. $\rho_i(\phi,\psi)=\rho_i(\psi,\phi)$ (symmetry).
3. If $\rho_i(\phi,\psi)=\rho_i(\phi_1,\psi_1)$ and $\rho_i(\phi_1,\psi_1)=\rho_i(\phi_2,\psi_2)$ then $\rho_i(\phi,\psi)=\rho_i(\phi_2,\psi_2)$ (transitivity equality).
4. $\rho_i(\phi,\psi)\le\rho_i(\phi,\chi)+\rho_i(\chi,\psi)$ (nonequality of a triangle).
5. $\phi\equiv\psi\Leftrightarrow\rho_i(\phi,\psi)=0$ ($\phi\equiv\psi$ here and further denotes equivalence of the formulas concerning to all models of the experts, that is for anyone expert i (assigning model $\mathbf{M}_i$) $\phi^{\mathbf{M}_i}=\psi^{\mathbf{M}_i}$ correctly).
6. $\phi\equiv\neg\psi\Rightarrow\rho_i(\phi,\psi)=1$.
7. $\rho_i(\phi,\psi)=1-\rho_i(\phi,\neg\psi)=\rho_i(\neg\phi,\neg\psi)$.
8. $\rho_i(\phi,\psi)=\rho_i(\phi\wedge\psi,\phi\vee\psi)$.
9. $\rho_i(\phi,\neg\phi)=\rho_i(\phi,\psi)+\rho_i(\psi,\neg\phi)$.

The proof of the theorem follows from definitions, properties of a probabilistic measure and logical evaluations.

The measure of refutability and their properties

From the point of view of importance of the information messaged by the expert it is natural to assume that the more above refutability (informativeness) of the formula, the smaller the number of models on which it is executed (the smaller measure of the set on which the formula is true). Therefore we shall enter the refutability as follows.

Definition 6. We call $I_i(P(\bar{x}))=\rho_i(P(\bar{x}),\ 1)$ the measure of refutability of formula $P(\bar{x})$, where **1** is the identical true predicate, that is $\bar{x}=\bar{x}$.

For entered distances obtained :

$$I_i(P)=\begin{cases}\dfrac{\sum\limits_{\mathbf{M}_i\in Mod_n(L)}\mu\left(\neg P^{\mathbf{M}_i}\right)}{\left|Mod_n(L)\right|}, & if\ \ \rho_1\\ \dfrac{\left|Mod\left(\neg P\right)\right|}{\left|Mod_n(L)\right|}, & if\ \ \rho_2\\ \dfrac{\min\limits_{\bar a\in M^{l(\bar a)}}\left|Mod\left(\neg P(\bar a)\right)\right|}{\left|Mod_n(L)\right|}, & if\ \ \rho_3\end{cases}$$

The following theorem is proved:

Theorem 2. For any formulas ("knowledge" of the experts) ϕ, ψ and anyone ρ_i on T the following assertions are valid:

1. $0\le I_i(\phi)\le 1$.
2. $I_i(1)=0$.
3. $I_i(0)=1$.
4. $I_i(\phi)=1-I_i(\neg\phi)$.
5. $I_i(\phi)\le I_i(\phi\wedge\psi)$.
6. $I_i(\phi)\ge I_i(\phi\vee\psi)$.
7. $I_i(\phi\wedge\psi)=\rho_i(\phi,\psi)+I_i(\phi\vee\psi)$.
8. If $\phi\equiv\psi$, then $I_i(\phi)=I_i(\psi)$.
9. If $\rho_i(\phi,\psi)=0$, then $I_i(\phi\wedge\psi)=I_i(\phi\vee\psi)=I_i(\phi)$.
10. $I_i(\phi\wedge\psi)=\dfrac{I_i(\phi)+I_i(\psi)+\rho_i(\phi,\psi)}{2}$.
11. $I_i(\phi\vee\psi)=\dfrac{I_i(\phi)+I_i(\psi)-\rho_i(\phi,\psi)}{2}$.

The entered definitions, formulated above the properties of the metric and logical evaluations are used for proof of this theorem (analog [2, 4]).

Bibliography (References)

1. Blohchisin V.Ya. and Lbov G.S. On Measure of Informativeness of Logical Statements // The Reports of Respubl. school – seminar "The Technology of Developing Expert Systems". Kishinev, 1978, p.12-14.
2. Vikent'ev A.A., Lbov G.S. Setting the metric and informativeness on statements of experts // Pattern Recognition And Image Analysis. 1997, v. 7 (2), p. 175-189.
3. Vikent'ev A.A., Lbov G.S. About Metrizations of the Boolean algebra of the Sentences and Informativenessof the Statements of the Experts // The Reports of RAS, 1998, t. 361 (2), p.174-176.
4. Vikent'ev A.A., Koreneva L.N. To a Problem on Distances between the Formulas which are Denoted the Structured Objects // Mathematical Methods of a Pattern Recognition .Moscow, 1999. P.151-154.
5. Chang C.C., Keisler H.J. Model Theory. Studies in Logic and Foundations of Mathematics. 1973, v. 73, 550 p.
6. Ershov Yu.L, Palyutin E.A. Mathematical Logic. Moscow, 1991, 336 p.
7. Gaifman H. Concerning measures in the first order calculus // Israel Journal of Mathematics, v. 2 (1), 1964, p. 1-18.
8. Fagin R. Probabilities on finite models. // The Journal of Symbolic Logic, v. 41 (1), 1976, p. 50-58.

Authors' Information

Alexandr A. Vikent'ev *– Senior Researcher, Ph. D, Institute of Mathematics, Siberian Division Russian Academy of Sciences, Laboratory Data Mining; Akad. Koptuga St., 4; Novosibirsk State University; vikent@math.nsc.ru*

ИЗМЕРИМЫЕ МОДЕЛИ ТЕОРИИ ПЕРВОГО ПОРЯДКА В РАССТОЯНИЯХ НА ВЫСКАЗЫВАНИЯХ И ВЕРОЯТНОСТЯХ НА ЗНАНИЯХ*

Александр Викентьев

***Аннотация**: При анализе знаний, заданных в виде высказываний экспертов, для различия содержащейся в них информации и группирования их по похожести, возникает необходимость введения расстояния между высказываниями экспертов и меры опровержимости (информативности) высказываний экспертов. Этой проблемой занимались Загоруйко Н.Г., Лбов Г.С., Викентьев А.А. [1-4]. Вводим расстояние на измеримых формулах с использованием не более чем счетных моделей некоторой, заранее фиксированной теории Т языка первого порядка. Такой подход является естественным при изучении некоторой конкретной прикладной проблемы, (поскольку тогда расстояние и информативность не будут искажены моделями, не относящимися к изучаемой аксиоматизированной области знаний) заданной например, некоторыми аксиомами-связями между переменными в ней, далее - теорией. Работа проделана в рамках проекта РФФИ 07-01-00331.*

***Keywords** : базы знаний,высказывания экспертов, теория моделй, метрика.*

***ACM Classification Keywords**: I.2.6. Artificial Intelligence - knowledge acquisition.*

Введение

К настоящему времени достаточно хорошо развиты теория и методы построения решающих функций распознавания образов на основе анализа эмпирической информации, представленной в виде таблиц данных. Параллельно этому проявляется все больший интерес к построению решающих функций на основе анализа экспертной информации, заданной в виде логических «знаний» нескольких экспертов, которые представлены формулами языка первого порядка [2-4]. При этом возникает задача введения расстояния на таких «знаниях» и определения мер опровержимости, информативности и вероятностей этих «знаний» [1-2] с учетом имеющейся теории Т и ее моделей.

При решении задач распознавания образов, кластерного и регрессионного анализа важную роль играет информация, полученная от экспертов. В трудноформализуемых областях исследований особую важность приобретают методы обработки эмпирической информации, представленной списком экспертных логических «знаний» (эти «знания» могут быть частично или полностью противоречивы) [1], но совместны с имеющейся теорией Т. Очевидно, такие высказывания («знания») могут различаться по количеству содержащейся в них информативности относительно Т.

Расстояние на высказываниях экспертов и его свойства

Рассмотрим сигнатуру $\Omega = \left\{ P_1^{n_1}, \ldots, P_f^{n_f} \right\}$, состоящую из конечного числа предикатных символов от конечного числа переменных, которые выбираются для записи и изучения имеющихся связей между переменными в конкретной прикладной области. Пусть выбрано некоторое исходное множество переменных $X = \{x_1, \ldots, x_k\}$, среди которых присутствуют переменные разных типов. Обозначим через D_{x_j} множество возможных значений переменной x_j. Пусть $A_n = \bigcup_{j=1}^{k} D_{x_j}$ - непустое множество

* Работа выполнена при поддержке РРФИ 07-01-00331 и Интеграция НГУ.

мощности не превосходящей некоторого числа n, являющееся объединением всех значений рассматриваемых переменных, включенных в предикаты.

Предполагаем, что в Ω для каждой переменной x_j включен сорт - одноместный предикат P_{x_j}, определенный только на области значений переменной x_j в A_n, то есть предикат $P_{x_j}(a)$ истинен в модели A_n на элементах $a \in A_n \Leftrightarrow a \in D_{x_j}$.

Во многих прикладных задачах значения переменных и их число конечны, и поэтому рассматриваются конечные множествам, но мы рассмотрим и более общий измеримый произвольный случай (модели могут быть и бесконечными: произвольной мощности n).

Определение1.[5,6]. Под интерпретацией будем понимать отображение γ, ставящее в соответствие каждому n_i- местному предикатному символу $P_i^{n_i}$ из сигнатуры Ω n_i- местный предикат (отношение) $P_i^{A_n} \subseteq A_n^{n_i}$, заданный на множестве A_n. Это позволяет говорить о модели $\langle A_n, \Omega \rangle$ сигнатуры Ω. В модели $\langle A_n, \Omega \rangle$ истинен предикат $P(x_1,\ldots,x_k)$ на элементах $a_1,\ldots a_k$ из A_n (записывается $\langle A_n, \Omega \rangle \models P(a_1,\ldots,a_k)$) тогда и только тогда, когда $\langle a_1,\ldots a_k \rangle \in P^{A_n}$ и $a_j \in D_{x_j}$. Ясно, что достаточно рассматривать модели только конечной сигнатуры, хотя результаты верны и в общей ситуации.

Пусть имеется конечное число s экспертов и области возможных значений всех переменных. Модели (в смысле теории моделей) задаются самими экспертами с условием , что они модели Т. Каждый эксперт задает свою модель. Каждый j-ый эксперт задает свою интерпретацию каждого предикатного символа $P_i^{n_i}$ сигнатуры Ω соответствующим отношением (предикатом) на множестве A_n. В результате имеем множество моделей $\{M_j\}_{j=1}^{s}$.

Считаем, что «Знания» экспертов можно записать в виде формул языка первого порядка, а формулы определяют формульные предикаты (подмножества) в каждой модели M_j по заданной интерпретации j-го эксперта [4].

Пусть F- система подмножеств множества $\mathrm{A} = \bigcup_k A_n^k$, где $A_n^k = \underbrace{A_n \times \ldots \times A_n}_{k \text{ раз}}$, образующая σ-алгебру. Нас будут интересовать только такие подмножества S_j из F, для которых найдется формула ψ_j, отражающая «знания» экспертов, которая и определяет это подмножество S_j (формульное подмножество). То есть S_j - это множество кортежей из A, на которых выполняется формула ψ_j. Формула ψ_j либо отражает какое-то из «знаний» экспертов, либо является их булевой комбинацией. В дальнейшем каждому рассматриваемому нами множеству S_j соответствует некоторая формула ψ_j.

Ω -множество всех используемых экспертами предикатных символов. Пусть B --замыкание множества Ω относительно логических операций $\neg, \vee, \wedge, \rightarrow$ и кванторов $\forall$ и $\exists$ по переменным. Ясно, что интересующее нас множество формул в B содержится.

Определение2. [8]. Вероятностной мерой μ на множестве B называется отображение $\mu: B \to [0,1]$, удовлетворяющее для ϕ и $\psi \in B$ условиям:

1) если $\vdash \phi \equiv \psi$, то $\mu(\phi) = \mu(\psi)$;

2) если $\vdash \phi$, то $\mu(\phi) = 1$;

3) если $\vdash \neg\phi$, то $\mu(\phi) = 0$;

4) $\mu(\neg\phi) = 1 - \mu(\phi)$;

5) если $\vdash \neg(\phi \wedge \psi)$, то $\mu(\phi \vee \psi) = \mu(\phi) + \mu(\psi)$.

Далее всегда предполагаем, что на множестве формул B задана вероятностная мера μ. Тем самым, вероятностная мера μ задана на множествах из F.

По заданным «знаниям» экспертов мы построили класс моделей (исходный класс). Чтобы более полно использовать информацию экспертов расширим исходный класс моделей теории Т. Учитывая одновременно информацию нескольких экспертов, можно доуточнить (расширить или сузить) каждую модель и эти новые модели теории Т добавить в исходный класс. Доуточнять модели можно следующим образом. Вместо «знания», заданного экспертом в виде предиката P_i в модели M_j, далее будем рассматривать его доуточнение - предикат $\widetilde{P}_i$.

Под доуточнением $\widetilde{P}_i$ предиката P_i понимаем другую интерпретацию этого предиката в модели M_j одним из способов:

1) оставить отношение без изменений;

2) исключить те элементы из P_i, в истинности которых j-ый эксперт не совсем уверен;

3) добавить в отношение новые элементы и исключить некоторые старые, например, с учетом «знаний» других экспертов;

4) выполнить пункты 2) и 3) одновременно.

Обозначим произвольный расширенный класс моделей теории Т через $Mod_n(\Omega)$. Введем расстояние на множестве «знаний» экспертов с помощью построенного класса моделей $Mod_n(\Omega)$ *мощности не более, чем n* . Модели различаются интерпретациями, но на них выполнены все аксиомы теории Т.

Определим расстояние между формульными подмножествами, (предикатами) в каждой модели $M_i \in Mod_n(\Omega)$, как меру их симметрической разности или ее модификацию.

Определение3. Расстоянием между различными предикатами $P_k^{M_i}$ и $P_j^{M_i}$, определенными в модели M_i, назовем величину

$$\rho_{M_i}\left(P_k^{M_i}, P_j^{M_i}\right) = \mu\left(\widetilde{P}_k^{M_i} \Delta \widetilde{P}_j^{M_i}\right).$$

Замечание. Это определение корректно, если предикаты одной местности и с одинаковым набором переменных. Если рассматриваемые предикаты имеют разную местность (арность) или разный набор переменных, и эксперт считает отсутствующую в одной из формул переменную x_i несущественной, полагаем, что она принимает любое из возможных значений. В противном случае (если она существенна) доопределяем эту переменную, добавив конъюнктивно к нужной формуле предикат $\widetilde{P}_{x_i}$, уточняющий значения этой переменной. В дальнейшем, с учетом выше сказанного, будем изучать расстояние между формулами от одних и тех же переменных (одной арности).

Можно рассматривать случай вычисления расстояния между различными интерпретациями одной и той же формулы на одном носителе, но это предмет другой работы.

Расстояние между формулами, определенными на множестве моделей $Mod_n(\Omega)$, определим как среднее на множестве расстояний в моделях.

Определение4. Расстоянием между формулами P_k и P_j, определенными на множестве $Mod_n(\Omega)$, назовем величину

$$\rho_1\left(P_k, P_j\right) = \frac{\sum\limits_{M_i \in Mod_n(\Omega)} \rho_{M_i}\left(P_k^{M_i}, P_j^{M_i}\right)}{\left|Mod_n(\Omega)\right|}.$$

Теперь рассмотрим способ определения расстояния между предложениями (замкнутыми формулами). Обозначим через $Mod(\phi)$ множество моделей из $Mod_n(\Omega)$, на которых истинно предложение ϕ, т.е. $Mod(\phi) = \left\{M_i \in Mod_n(\Omega) \mid \ M_i \models \phi\right\}$.

Очевидно, существуют такие модели, на которых предложение истинно, и такие, на которых оно ложно (если оно не тавтология). Естественно измерять различие информации, содержащейся в предложениях, количеством моделей, на которых предложения принимают разные значения истинности.

Определение5. Расстоянием между предложениями ϕ и ψ назовем величину

$$\rho_2\left(\phi, \psi\right) = \frac{\left|Mod\left(\left(\phi \wedge \neg\psi\right) \vee \left(\neg\phi \wedge \psi\right)\right)\right|}{\left|Mod_n(\Omega)\right|}.$$

Рассмотрим еще один способ определения расстояния между формулами. Дополним множество Ω константами из множества носителей моделей. Для этого множества M рассмотрим произвольные кортежи $\overline{a}$ длины местности формул, равной $l(\overline{a})$. При подстановке кортежей в формулы, в предположении, что формулы имеют одинаковую местность (как этого добиться, было указано выше), формулы становятся предложениями.

Определение6. Расстоянием между формулами $\phi(\overline{x})$ и $\psi(\overline{x})$ назовем величину

$$\rho_3\left(\phi(\overline{x}), \psi(\overline{x})\right) = \min_{\overline{a} \in M^{l(\overline{a})}} \rho_2\left(\phi(\overline{a}), \psi(\overline{a})\right).$$

Доказана следующая теорема, из которой следует, что предложенные расстояния действительно являются метриками. В теореме приведены и некоторые дополнительные свойства введенных расстояний. При доказательствах теорем используется теоретико модельные методы цитируемых работ [1-8].

Теорема1. Для любого измеримого расширения (= все интересующие нас формулы-знания имеют измеримую область истинности, т.е. меру) исходного класса моделей теории T для любых формул («знаний» экспертов) ϕ, ψ, χ и для любой функции ρ_i справедливы следующие свойства:

1. $0 \le \rho_i(\phi,\psi) \le 1$
2. $\rho_i(\phi,\psi) = \rho_i(\psi,\phi)$ (симметричность расстояния).
3. $\rho_i(\phi,\psi) \le \rho_i(\phi,\chi) + \rho_i(\chi,\psi)$ (неравенство треугольника).
4. $\phi \equiv \psi \Leftrightarrow \rho_i(\phi,\psi) = 0$ ($\phi \equiv \psi$ здесь и далее обозначает эквивалентность формул относительно всех моделей, то есть для любой модели M_i верно $\phi^{M_i} \equiv \psi^{M_i}$)
5. $\phi \equiv \neg\psi \Rightarrow \rho_i(\phi,\psi) = 1$
6. $\rho_i(\phi,\psi) = 1 - \rho_i(\phi,\neg\psi) = \rho_i(\neg\phi,\neg\psi)$.
7. $\rho_i(\phi,\psi) = \rho_i(\phi \wedge \psi, \phi \vee \psi)$.
8. $\rho_i(\phi,\neg\phi) = \rho_i(\phi,\psi) + \rho_i(\psi,\neg\phi)$.

Доказательство теоремы следует из определений, свойств вероятностной меры, теоретико-модельных и логических вычислений как в [7].

Мера опровержимости (информативности)высказываний экспертов, вероятности формул и их свойства

С точки зрения важности информации, сообщенной экспертом, естественно считать, что информативность высказывания (непустого предиката) тем выше, чем меньше число элементов, ему удовлетворяющих (более точно, чем меньше мера, определенная на этом подмножестве). Поэтому введем меру опровержимости, которая используется как информативность для выполнимых формул, следующим образом.

Определение7. Пусть $\phi(\bar{x})$ - формула, отражающая «знание» эксперта, тогда мерой опровержимости формулы $\phi(\bar{x})$ назовем величину $I_i(\phi(\bar{x})) = \rho_i(\phi(\bar{x}),1)$, где **1** -- тождественно истинный предикат, например, $\bar{x} = \bar{x}$.

Для введенных расстояний получаем:

$$I_i(\phi) = \begin{cases} \dfrac{\sum_{M_i \in Mod_n(\Omega)} \mu\left(\neg\phi^{M_i}\right)}{\left|Mod_n(\Omega)\right|}, \text{если } \rho_1 \\ \dfrac{\left|Mod(\neg\phi)\right|}{\left|Mod_n(\Omega)\right|}, \text{если } \rho_2 \\ \dfrac{\min_{\bar{a} \in M^{l(\bar{a})}} \left|Mod(\neg\phi(\bar{a}))\right|}{\left|Mod_n(\Omega)\right|}, \text{если } \rho_3 \end{cases}$$

Для меры опровержимости справедлива следующая теорема.

Теорема2. Для любого измеримого расширения исходного класса моделей теории T для любых формул ϕ, ψ и для любой функции ρ_i справедливы следующие утверждения:

1. $0 \le I_i(\phi) \le 1$.
2. Если $\phi \equiv \psi$, то $I_i(\phi) = I_i(\psi)$.
3. $I_i(1) = 0$ (1 - тождественно истинная формула)
4. $I_i(0) = 1$ (0 - тождественно ложная формула).
5. $I_i(\phi) = 1 - I_i(\neg\phi)$.
6. $I_i(\phi) \le I_i(\phi \wedge \psi)$.
7. $I_i(\phi) \ge I_i(\phi \vee \psi)$.
8. $I_i(\phi \wedge \psi) = \rho_i(\phi, \psi) + I_i(\phi \vee \psi)$.
9. Если $\rho_i(\phi, \psi) = 0$, то $I_i(\phi \wedge \psi) = I_i(\phi \vee \psi) = I_i(\phi)$.
10. $I_i(\phi \wedge \psi) = \dfrac{I_i(\phi) + I_i(\psi) + \rho_i(\phi, \psi)}{2}$.
11. $I_i(\phi \vee \psi) = \dfrac{I_i(\phi) + I_i(\psi) - \rho_i(\phi, \psi)}{2}$.

Для доказательства теоремы используются введенные определения, сформулированные выше свойства метрики и теоретико-модельные вычисления.

На практике же чаще эксперт задает высказывание с его «вероятностью». А вопрос состоит в изучении и согласовании таких высказываний, введении некоторой метрики на таких высказываниях. Первоочередной задачей, на наш взгляд, является определение вероятностей для формул с помощью модельного подхода.

Определение 8. Пусть $\phi(\bar{x})$ - формула, выражающая знание эксперта, тогда вероятность высказывания эксперта определим как величину $P_i\left(\phi(\bar{x})\right) = I_i\left(\neg\phi(\bar{x})\right) = \rho_i\left(\phi(\bar{x}), \neg 1\right)$.

Теорема3. Для любого измеримого расширения исходного класса моделей теории T для любых формул ϕ, ψ и для любого ρ_i справедливы следующие утверждения:

1- $0 \le P_i(\phi) \le 1$.
2. Если $\phi \equiv \psi$, то $P_i(\phi) = P_i(\psi)$.
3. $P_i(1) = 1$.
4. $P_i(0) = 0$.
5. $P_i(\phi) = 1 - P_i(\neg\phi)$.
6. $P_i(\phi) \ge P_i(\phi \wedge \psi)$.

7. $P_i(\phi) \le P_i(\phi \vee \psi)$.

8. $P_i(\phi \wedge \psi) = P_i(\phi \vee \psi) - \rho_i(\phi, \psi)$.

9- Если $\rho_i(\phi, \psi) = 0$, то $P_i(\phi \wedge \psi) = P_i(\phi \vee \psi) = P_i(\phi)$.

10. $P_i(\phi \wedge \psi) = \dfrac{P_i(\phi) + P_i(\psi) - \rho_i(\phi, \psi)}{2}$.

11. $P_i(\phi \vee \psi) = \dfrac{P_i(\phi) + P_i(\psi) + \rho_i(\phi, \psi)}{2}$.

Так вычисленные с использованием модельного подхода вероятности позволяют уточнять «вероятности» экспертов с учетом моделей теории Т и будут применяться в дальнейшем.

Заключение

Полученные результаты остаются справедливыми для произвольных моделей и формульных подмножеств, аппроксимируемых измеримыми и отвечающим знаниям экспертов. Результаты можно использовать для нахождения усредненных расстояний, мер опровержимости и вероятностей высказываний экспертов. Исследование найдет применение к вопросам наилучшего согласования экспертных высказываний, построения решающих функций распознавания образов и разработки экспертных систем.

Список литературы

1. Г.С. Лбов, Н.Г. Старцева. Логические решающие функции и вопросы статистической устойчивости решений. Новосибирск: Издательство Института математики, 1999. С. 85-102.
2. Н.Г. Загоруйко, М.В. Бушуев. Меры расстояния в пространстве знаний // Анализ данных в экспертных системах. Новосибирск,1986. Выпуск 117:Вычислительные системы. С.24-35.
3. А.А. Викентьев, Г.С. Лбов. О метризации булевой алгебры предложений и информативности высказываний экспертов // Доклад РАН 1998.Т.361, №2 С.174-176.
4. A.A. Vikentiev, G.S. Lbov. Setting the metric and informativeness on statements of experts // Pattern Recognition and Image Analysis. 1997 V.7, N2, P.175-183.
5. Г. Кейслер, Ч.Ч. Чэн Теория моделей. Москва:Мир,1977.
6. Ю.Л. Ершов, Е.А. Палютин Математическая логика. Санкт-Петербург, 2004.
7. А.А.Викентьев, Л.Н. Коренева К вопросу о расстояниях между формулами, описывающими структурированные объекты. // Математические методы распознавания образов (ММРО-99). РАН ВЦ, Москва, 1999. С.151-154.
8. H. Gaifman Concerning measures in the first order calculus. // Israel Journal of Mathematics, v. 2 (1), 1964, p. 1-18.

Информация об авторе

Александр А. Викентьев *– с.н.с., канд.физ-мат.наук , Институт математики СО РАН, пр. Академика Коптюга, д.4, Лаборатория анализа данных. Доцент, Новосибирский госуд. университет; e-mail: vikent@math.nsc.ru*

ИНФОРМАЦИОННАЯ МОДЕЛЬ ОБРАБОТКИ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ТЕКСТОВ

Александр Палагин, Виктор Гладун, Николай Петренко, Виталий Величко, Алексей Севрук, Андрей Михайлюк

Аннотация: *В статье рассматривается формальная модель обработки естественно-языковых текстов в знаниеориентированных информационных системах. Описаны компоненты, реализующие функции предложенной формальной модели.*

Ключевые слова: *обработка естественно-языковых текстов.*

Архитектура современных знаниеориентированных информационных систем (ЗОС) с естественно-языковым представлением и обработкой знаний включает онтологическую составляющую эксплицитно, которую в общем виде можно интерпретировать как концептуальную базу знаний. Такая база знаний представляется в виде ориентированного графа, вершинами которого являются фреймы, описывающие концепты, а дугами – множество концептуальных отношений, связывающих между собой концепты. Другой важной особенностью указанной архитектуры является разделение и отдельная обработка семантики первой и второй ступени [1], что в общем случае означает разделение внутриязыкового и внеязыкового процессинга [2] и переход к формально-логическому представлению исходного текста.

Указанные особенности архитектуры современных ЗОС трансформируют традиционную модель обработки естественно-языковых текстов (ЕЯТ) в формальную модель следующего вида

$F = \langle T, W, SS^1, O, S^2, I \rangle$, где

T – множество обрабатываемых ЕЯТ;

W – множество словоформ, входящих в T;

SS^1 – множество синтактико-семантических структур первой ступени, описывающих T;

O – множество онтологических структур, отображающих множества W и SS^1 в S^2;

S^2 – множество семантических структур второй ступени, описывающих множество сценариев T;

I – множество информационно-кодовых представлений S^2.

Опишем объекты формальной модели.

Множество T представляет совокупность естественно-языковых текстов, характеризующихся стилями делового и научно-технического характера.

Цепочка $W \to SS^1$ в классическом понимании представляет грамматический анализ ЕЯТ. В отличие от традиционных линейного и сильнокодированного методов анализа мы используем смешанный метод анализа. Суть его состоит в том, что в лексикографической базе данных полное множество W представлено в таблицах двух типов: таблицами лексем с соответствующими морфологическими, синтаксическими и семантическими характеристиками и таблицами флексий для всех полнозначных, изменяющихся частей речи. При этом алгоритмы формирования парадигмы лексем просты; в таблицах

лексики указаны основы лексем и соответствующие коды для выбора записей из таблиц флексий. Нефлексийные изменения учитываются соответствующими алгоритмами.

Описанная структура грамматического анализа однозначно соответствует эффективному отображению функциональных операторов на аппаратный уровень реализации, в частности в базисе ПЛИС (программируемые логические интегральные схемы).

Множество O онтологических структур в идеале представляет языково-онтологическую картину мира, описанную в [1, 3].

Множество SS^1 формируется и интерпретируются итерационно подсистемой синтактико-семантического анализа типа “Конспект” [4]. Основной операцией синтактико-семантического анализа является распознавание синтаксических и семантических отношений, связывающих слова текста. Распознавание связей между знаменательными словами осуществляется путем анализа флексий и предлогов на основе лексических моделей без использования в явном виде правил традиционной грамматики. Для каждого предложения исходного текста строится дерево разбора. Разрешение семантической неоднозначности осуществляется путем обращения к множеству онтологических структур O. На основе построенных деревьев разбора фраз строится категориальная сеть, представляющая собой семантическое пространство S^2 текста. В качестве компьютерного представления такого пространства текста удобно использовать растущую семантическую сеть множества информационно-кодовых представлений I, организованную на основе пирамидальной сети, рецепторы которой соответствуют именам объектов, классов объектов, свойств, состояний, действий, отношений, семантических падежей, модификаторов [5].

Цепочка преобразования информации $T \to W \to SS^1$ и $O \to S^2 \to I$, по сути, представляют (соответственно) базовые процедуры анализа и понимания ЕЯТ, средствами интерпретации которых являются грамматический и семантический процессоры.

В приложениях поиска и обработки большого объема текстовых документов целесообразно использовать знаниеориентированную поисковую систему [6], обеспечивающую начальный и конечный этапы обработки документов - поиска в Интернет и сохранения документов в базе данных в виде их конспектов, сгенерированных подсистемой “Конспект”.

Описанная модель обработки естественно-языковых текстов в знаниеориентированной информационной системе, включающей подсистему “Конспект” как компоненту, представляет перспективное направление развития онтолого-управляемых информационных систем, активно использующих онтологию лексики естественного языка.

Литература

Палагін О.В., Петренко М.Г. Модель категоріального рівня мовно-онтологічної картини світу //Математичні машини і системи. – 2006. - №3. - С.91-104.

Рубашкин В.Ш. Представление и анализ смысла в интеллектуальных информационных системах. – М.: Наука, 1989. – 191с.

Палагин А.В. Организация и функции "языковой" картины мира в смысловой интерпретации ЕЯ - сообщений //Information Theories and Application. – 2000. – Vol. 7, №4. С.155-163.

Гладун В.П., Величко В.Ю. Конспектирование естественно-языковых текстов. Proceedings of the XI-th International Conference “Knowledge-Dialogue-Solution”(KDS’2005).- Varna, Bulgaria.-2005.- pp.344-347 vol.2.

Гладун В.П. Планирование решений. - Киев: Наукова думка, 1987. -168с.

Севрук О.О., Петренко М.Г. Знання-орієнтована пошукова система на основі мовно-онтологічної картини світу //Тези доповідей XIII міжнародної конференції з автоматичного управління “Автоматика-2006”. – Вінниця. – 2006. - 25-28 вересня. – С.413.

Информация об авторах

Палагин Александр Васильевич *- Ин-т кибернетики им. В.М. Глушкова НАН Украины, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40,e-mail: palagin_a@ukr.net*

Гладун Виктор Поликарпович *- Ин-т кибернетики им. В.М. Глушкова НАН Украины, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40,e-mail: glad@aduis.kiev.ua*

Петренко Николай Григорьевич *- Ин-т кибернетики им. В.М. Глушкова НАН Украины, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40,e-mail: petrng@ukr.net*

Величко Виталий Юрьевич *- Ин-т кибернетики им. В.М. Глушкова НАН Украины, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40,e-mail: glad@aduis.kiev.ua*

Севрук Алексей Олегович, Михайлюк Андрей Васильевич *- Ин-т кибернетики им. В.М. Глушкова НАН Украины, Киев-187 ГСП, 03680, просп. акад. Глушкова, 40,e-mail: petrng@ukr.net*

KNOWLEDGE-BASED APPROACH TO DOCUMENT ANALYSIS

Elena Sidorova, Yury Zagorulko, Irina Kononenko

Abstract*: The paper presents an approach to extraction of facts from texts of documents. This approach is based on using knowledge about the subject domain, specialized dictionary and the schemes of facts that describe fact structures taking into consideration both semantic and syntactic compatibility of elements of facts. Actually extracted facts combine into one structure the dictionary lexical objects found in the text and match them against concepts of subject domain ontology.*

Keywords*: text analysis, ontology, natural language, fact extraction.*

ACM Classification Keywords*: I.2.7 Natural Language Processing - Text analysis*

Introduction

The development of information systems such as intellectual document management systems or knowledge portals is one of the most actual tasks for today. This task is often considered within the framework of creating the systematized document storehouses to simplify the search for the necessary information. Despite the importance of these questions the opportunities provided by existing information systems appear to be insufficient for the intellectual organization of activity: first, it becomes difficult (practically impossible) to find the necessary information in constantly expanding archive; second, the data are often duplicated and contradict each other.

Modern information systems should be capable to solve the whole complex of tasks concerned with the management of a stream of ingoing «crude data», namely automatic classification and automatic indexing of texts, operative and adequate document routing, data transmission, storage, archiving and content -based search.

The technology is developed to automatically analyse texts of business or scientific documents in information system operating within restricted subject domains. It should provide correct addition of new documents in information space of the system and support the content-based search in it. This technology have to support adjustment of the knowledge base of the information system both in the process of its creation and during its operation [Kononenko et al., 2005].

Knowledge and data representation

The technology of text analysis uses three components of knowledge:

- ontology that includes concepts and relations of subject domain; from the point of view of the analysis the ontology describes data to be extracted from texts and placed in the database of the system;
- dictionary (thesaurus) that contains terms that represent concepts and relations of the ontology in texts;
- information content of the system, or a database.

In the system data are presented as a set of information objects (IO) of various types that describe objects of the subject domain and, in the aggregate, form information content of the system. Each IO is an instance of some element of the ontology (concept or relation) and has the structure with the fixed set of attributes specified by the expert.

Any IO may be considered as having three different aspects - structure, content, and context. The structure is characterized by a set of own attributes and attribute values. The context specifies possible environment of IO and is defined by a set of relations with other information objects. The format of IO structure and context is defined by ontology.

For example, the context of IO can be formed by the following relations of the ontology:

Part (Publication, Collection) - the relation that connects a portion to the whole (e.g., an article and a collection of articles);

Author (Person, Document) - the relation that connects a document and a creator of the document;

Publisher (Organization, Collection) - the relation that connects the book with the organization that issues it.

Besides descriptions of objects of the subject domain, the information system also contains information objects that represent various information resources, such as publication, Internet page, diagram, map, etc. The content of such resources is described by a network of the domain objects.

The technology of the analysis is aimed at processing of text information resources. Below such IOs are named *documents*.

To provide the analysis of the document text we have to perform the following actions:

- specify concept (classes) of documents and insert them in ontology;
- define the formal structure of the text for each class of documents;
- describe the schemes of the facts setting rules of extraction of facts from the text.

Formal structure of the text

In the proposed approach documents to be analyzed are information objects that are described by a certain concept (class) of the ontology, for example, *Document* class. The text representing the contents of objects of the Document class (or any other class describing a text resource) is analyzed with the purpose of extraction of the significant information, or content. The content of the document includes a set of information objects and their relations extracted from the document text.

The formal structure of the text depending on a type or genre of the document is used In the process of document analysis.

According to [Zhigalov et al., 2002] text in the digital form has at least three levels of formal structure, i.e. physical, logic and genre levels. The first one concerns presentation of the text on page, for example, by means of tags or tables of styles. The second level concerns such elements as text, paragraph, line, sentence, etc. The third level is presented by decomposition of text into genre parts. For example, the text of the business letter [Kononenko et al., 2002] includes the following genre sections: heading (sender, addressee, resume, and address), basic section (text of the letter, comments and enclosure notice), and signature.

Below any formal text structure is named as a *segment* and described by markers. The marker is defined by the list of alternative elements where an element can be:

1) a symbol or a string;

2) lexical object identified in the process of lexical analysis;

3) segment of other type.

A segment is constructed starting from following restrictions:

- *single* - the segment should not intersect with other segments of the same type; a special case of this restriction is requirement of the absence of nesting of segments;
- *min* - segment must be minimal one in the given section of text;
- *max* - segment must be maximal one in the given section of text.

The scheme of the fact

Hierarchies of classes of concepts and semantic relations defined in ontology allow one to present structure of the proposition from a subject domain in form of a fact. A set of facts constitutes propositional content of the document.

In the proposed approach the analysis is aimed at extraction of only those facts that include objects and relations of the given subject domain. The declarative description of structure of the fact and conditions (restrictions) of its extraction are named *the scheme of the fact*.

The scheme of the fact includes a set of arguments (we use only unary and binary facts) where argument can be:

- concept of ontology;
- object or class of the dictionary;
- type of the fact;
- IO of the document whose text is being analyzed.

The scheme of the fact also includes description of restrictions which are imposed on compatibility of arguments. There are semantic and structural type restrictions.

From the point of view of the result the dynamic and static schemes are distinguished. A new object (IO or the fact) is created as a result of applying the dynamic scheme; the appearance of new object can serve as a basis for application of another scheme etc. Application of the static scheme leads to changing one of the arguments, for example, IO of the document or existing object. Generally, in the course of text analysis a set of objects or relations found in the given section of text is formed.

Let us to give examples of schemes of facts:

F1: Research-Object (monument) + Locality (Western Sahara) => creation
Object-is found-in (monument, Western Sahara)

F2: Activity (work) + Object (construction project) => creation

Function (work, construction project, Kind_of_Activity: construction)

F3: Sender (Organization) + Function.Kind_of_Activity => editing

Document (Kind_of_Activity: Function.Kind_of_Activity)

Semantic restrictions

Semantic restriction is imposed on semantic characteristics of arguments of the fact. Restriction explicitly presents a pair of compatible components, where a component is a class, or a dictionary term, or the values of attribute.

For each scheme of the fact the table of semantic combinations can be generated. The table should be filled by the expert. This table is applied for:

- narrowing the set of variants of possible combinations of text units;
- accounting for mutual influence of arguments (i.e., specification of a semantic class);
- specifying attributes of resulting object.

Below we can see a little fragment of the table of semantic combinations:

Work (class) + Construction_project (class) => Work: construction

"Development" (term) + Natural_resources (class) => Work: nature management

"Development" (term) + Document (class) => Work: document creation

Structural restrictions

Besides semantic restrictions, restrictions of other language levels, such as syntactic and genre restrictions, must be considered.

For each scheme of the fact additional conditions on its arguments should be given:

- a condition on a segment, i.e. what type of a segment the arguments should be discovered within;
- position of arguments in the text (contact position, pre- and postposition, priority of positions in case of multiple choice);
- syntactic conditions (valences of terms, prepositional phrases, etc.);
- rules of combining (coordination, projectivity, maximal connectivity).

Verification of syntactic compatibility may involve simple comparison of syntactic features of terms or construction of a local syntactic dependency tree [4].

Consider an example of scheme of the fact with structural restrictions:

Fact (a1:Work, a2:Object)

- condition on a Sentence segment:;
- check valences of terms of Work class;
- check syntactic compatibility;
- search for coordinated terms;
- conform to projectivity rule;
- give priority to the postposition of Object terms relative to Work terms.

Apply this scheme to following sentence:

“It takes about 2 months to complete the installation <1> of equipments <2> and systems of automatics <3> in view of the necessary field change <4>, carrying out of production tests <5> and preparations for shipment <6> of the 2-nd diesel power stations <7>.”

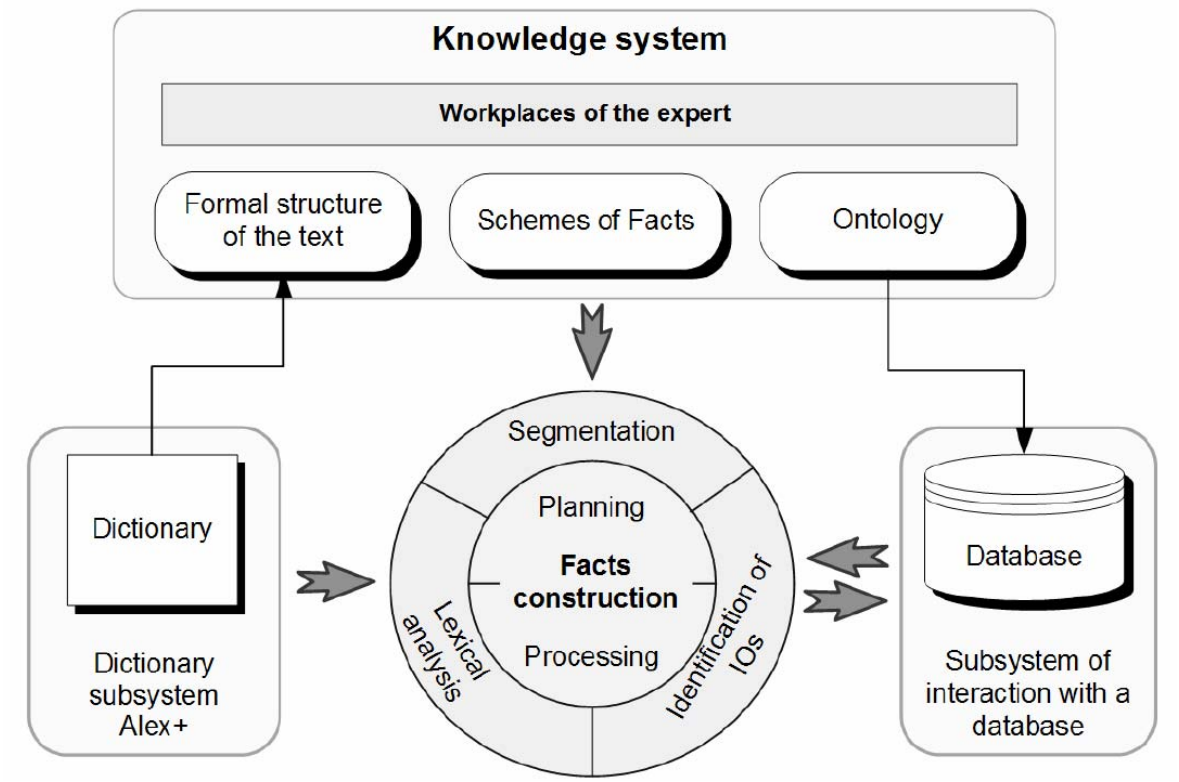
The following facts have been extracted from this sentence:

1. <1> [installation] - <2> [equipment]
2. <1> [installation] - <3> [systems of automatics y]
3. <4> [field change] - <2> [equipment]
4. <4> [field change] - <3> [systems of automatics]
5. <5> [production tests] - <2> [equipment]
6. <5> [production tests] - <3> [systems of automatics]
7. <5> [production tests] - <7> [power station]
8. <6> [shipment] - <7> [power station]

Technology of the text analysis

The system architecture (see Fig.1) includes four basic components: kernel, dictionary subsystem, editors of knowledge (ontology, schemes of facts, formal structures of the text), and a subsystem of interaction with a database.

The kernel of the system provides extraction of facts in accordance with the descriptions created by editors. The dictionary subsystem [Sidorova, 2005] ensures creation of the dictionary and realization of preliminary stage of text processing (segmentation, lexical and morphological analysis). The components realized within the project on creation of knowledge portals [Borovikova et al., 2005] are used as an editor of ontology and a module of interaction with a database.



Pic. 1. The architecture of system of the analysis.

Segmentation

There are two kinds of text segmentation - primary and genre ones.

During primary segmentation splitting linear representation of the text into ordered list of the string objects which are used for forming segments is carried out.

Genre segmentation is performed after the lexical analysis. It is based on lexical objects that mark different genre segments.

The mechanism of segmentation is realized by the Alex system [Zhigalov et al., 2002] included in the dictionary component of technology.

The lexical analysis

The lexical analysis performs extraction of lexical objects from the set of the ordered string objects obtained by the primary segmentation of the text. Lexical object is either a lexical pattern described in the Alex system, or a word/phrase represented in the dictionary.

The tasks of the given stage are following:

- application of lexical patterns;
- execution of the morphological analysis and phrase search;
- identification of genre segments.

The process results in the ordered list of objects with a following set of parameters: name (canonical form of a word or phrase, name of a pattern), position in the text, value (the main word in a synonymic group, numerical value, etc.), grammatical class (morphosyntactic information about the word form), semantic class, statistical characteristics.

Constructing facts

The mechanism of constructing facts is based on preliminary planning which is performed for each class of documents on the basis of the pre-specified schemes of facts.

Tasks of planning are the following:

1) Generation of executed rules on the basis of schemes of facts.

2) Organization of queue of rules to be executed. On this subject two aspects are taken into account:

- interdependence of schemes of facts and the order of creating objects;
- order of segments and their nesting level (the analysis is carried out starting with the smallest segment in the nesting hierarchy and proceeded up to the largest one).

3) Maintenance of correctness and convergence of process of fact construction.

During the document processing the rules are successively taken from the queue and applied. This process goes on until the queue becomes empty. For each rule data are grouped around the segments specified in a condition of a rule. Extraction of the facts is limited by frameworks of one segment.

The table of semantic combinations and syntactic rules (serving for checking of compatibility of grammatical characteristics of terms and controlling of coordination, projectivity, connectivity) are also used for fact construction.

For list of lexical objects obtained after lexical analysis the appropriate combinations are selected from the table of semantic combinations. These combinations are further considered as separate schemes of facts (however, syntactic rules are to be applied as well).

The closely adjacent objects of the same class are combined in one group. After that the contact groups are checked for compatibility (semantic and syntactic).

All the methods use the same approach to disambiguation that is based on use of weights of terms and objects. The weight depend on the following factors:

- term being a part of a phrase;
- compatibility of adjacent terms;
- term being a constituent of a fact;
- statistical characteristics, etc.

Identification of information objects

The further processing consists in forming of content of the document. For this purpose it is necessary to identify the obtained objects and provide their correct insertion into information space of the system.

The tasks of the given stage are as follows:

- Reconstruction of objects with complex structural names by means of use of “part-whole” hierarchy determined in a database;
- Reference resolution (identical objects are integrated);
- Search in a database for the objects found in the text of the document;
- Disambiguation, in case when the database includes several objects the description (content) of that corresponds to the obtained object.

The object is considered as *identified* if its class and a set of its key attributes are defined. This property allows us to distinguish the obtained object from other objects, i.e. uniqueness of objects in a database of the system is ensured.

The set of unambiguously identified objects forms a content of the document. Uniqueness of objects in the content provides its correct insertion into database of system.

Conclusion

The proposed approach is substantially based on ideas presented in [Narin'yani, 2002], in particular, we exploited idea of collaborative use of subject domain ontology and thesaurus as well as methods of semantically oriented analysis of text. In the course of practical implementation of proposed approach were also used methods and algorithms developed for experimental system for information extraction from weather forecast telegrams [Kononenko et al., 2000] and industrial intelligent document management system InDoc [Zagorulko et al., 2005].

Our immediate goals are to complete a creation of technology based on proposed approach and to apply it to solution of the laborious problem concerned with a filling of a knowledge portal with new knowledge and data [Borovikova et al., 2005].

Bibliography

[Borovikova et al., 2005] Borovikova O., Bulgakov S., Sidorova E., Zagorulko Yu. Ontology-based approach to development of adjustable knowledge internet portal for support of research activity // Bull. of NCC. Ser.: Comput. Sci. 2005. Is. 23, pp. 45-56.

[Gershenzon et al., 2005] Gershenzon., Nozhov I., Pankratov. Century System of extraction and search of structured information in big media text collections. Architectural and linguistic features. // Works of the international conference Dialogue'2005 "Computer linguistics and intellectual technologies". M.:Science, 2005, pp. 97-101. (in russian)

[Kononenko et al., 2000] Kononenko I., Kononenko S., Popov I., Zagorul'ko Yu. Information Extraction from Non-Segmented Text (on the material of weather forecast telegrams). // Content-Based Multimedia Information Access. RIAO'2000 Conference Proceedings, v.2, 2000, pp.1069-1088.

[Kononenko et al., 2002] Kononenko I.S., Sidorov E.A. Business letter processing as a part of documents circulation system // Works of the international seminar Dialogue'2002 on computer linguistics and its applications. M.:Science, 2002. V.2, pp. 299-310. (in russian)

[Narin'yani, 2002] A.S. Narin'yani. TEON-2: from Thesaurus to Ontology and backwards // The international seminar Dialogue'2002 on computer linguistics and its applications. M.:Science, 2002. V.1, pp. 199-54. (in russian)

[Sidorova, 2005] Sidorova E. Technology of development of thematic dictionaries based on a combination of linguistic and statistical methods // The international conference Dialogue'2005 "Computer linguistics and intellectual technologies". M.:Science, 2005, pp.443-449. (in russian)

[Zagorulko et al., 2005] Zagorulko Yu., Kononenko I., Sidorova E. A Knowledge-based Approach to Intelligent Document Management // CSIT'2005. Ufa-Assy, Russia, 2005. V1, pp. 33-38.

[Zhigalov et al., 2002] Zhigalov Vlad, Zhigalov Dmitrij, Zhukov Alexandre, Kononenko Irina, Sokolova Elena, Toldova Svetlana. ALEX - a system for multi-purpose automatized text processing // The international seminar Dialogue'2002 on computer linguistics and its applications. M.:Science, 2002. V.2, pp.192-208. (in russian)

Authors' Information

Elena Sidorova - *A.P. Ershov Institute of Informatics Systems; P.O.Box: pr. Lavrent'eva, 6, Novosibirsk, Russia, 630090; e-mail: lena@iis.nsk.su*

Yury Zagorulko - *A.P. Ershov Institute of Informatics Systems, P.O.Box: pr. Lavrent'eva, 6, Novosibirsk, Russia, 630090; e-mail: zagor@iis.nsk.su*

Irina Kononenko - *A.P. Ershov Institute of Informatics Systems; P.O.Box: pr. Lavrent'eva, 6, Novosibirsk, Russia, 630090; e-mail: irina_k@cn.ru*

AUTOMATED RESPONSE TO QUERY SYSTEM

Vladimir Lovitskii, Michael Thrasher, David Traynor

Abstract: *SMS (Short Message Service) is now a hugely popular and a very powerful business communication technology for mobile phones. In order to respond correctly to a free form factual question given a large collection of texts, one needs to understand the question at a level that allows determining some of constraints the question imposes on a possible answer. These constraints may include a semantic classification of the sought after answer and may even suggest using different strategies when looking for and verifying a candidate answer. In this paper we focus on various attempts to overcome the major contradiction: the technical limitations of the SMS standard, and the huge number of found information for a possible answer.*

Keywords: *mobile text messages, text message analysis and question-answering system*

ACM Classification Keywords: *I.2 Artificial intelligence: I.2.7 Natural Language Processing: Text analysis.*

Introduction

This paper represents results of our further research in the text data mining and the natural language processing areas [1-5] restricted by mobile's text-based SMS messaging. SMS is now a very powerful business communication technology widely used from small businesses and home users through to large corporations, governmental and non-governmental organisations. However, many of these users have little or no experience of SMS technology and only a vague idea of how successful they could be when properly harnessing the power of SMS communication.

SMS text messaging is currently being evaluated in many different areas:

- **Mobile Banking.** Mobile Banking Services including: Account Balance Enquiry, Account Statement Enquiries, Cheque Status Enquiry, Cheque Book Requests, Fund Transfer between Accounts Credit/Debit Alerts, Minimum Balance Alerts, Bill Payment Alerts, Bill Payment, Recent Transaction History Requests, Information Requests i.e. Interest Rates/Exchange Rates. (e.g. the HSBC's SMS Enquiry Service [6]). Although these services are appearing they do raise specific issues concerning security, especially when the data is extremely sensitive and confidential. 2ergo have recognised this and have developed and patented a secure messaging protocol [7] to address such issues through encryption technology.
- **Mobile Government** services [8]. For example, Ireland's tax collection agency, Office of the Revenue Commissioners, as of 2005 now receives at least as many enquiries by text message as by telephone. The SMS enquiry service allows citizens to claim tax credits and request a number of tax forms and information leaflets by sending text messages from their mobile phones.
- **Mobile Learning** (SMS in education [9]). Students find SMS messages useful and are keen to use it in a number of ways: announcements, assessment marks, assessment feedback, appointments, revision tips and also quick easy access to some library service. Thus, it is believed that SMS in education can facilitate collaboration, strengthen community spirit, assist timely completion of coursework and other assessments, and ultimately reduce the attrition rate amongst students. 2ergo have released a new product targeted at the education market [7]. It is aimed at anti-truancy, anti-bullying and group messaging to provide school administrators with the means to manage and control these very important issues.
- **Mobile Airport** services [10]. The SMS service gives passengers fast responses to queries about the airport when they are possibly in transit and scheduled to connect with a flight. It reduces the need for people to phone the airport information number, which can often be engaged and thereby increasing anxiety levels amongst passengers, or go to the information desk inside the terminal.

Given the recent experience the list of areas where mobile messaging will prove useful is likely to be extended. What these applications have in common is that they represent a **W**ell-**D**efined **A**pplication **D**omain (WDAD). For each of them a 5-digit short code can be predefined, e.g. when the message is sent to the 64222 short code WDAD "Edinburgh Airport" will be selected automatically and therefore the answer to the question *"Where is the nearest hotel?"* will be synthesized relatively simply by the Question-Answering System (QAS).

The central question to be addressed by this paper, however, is how to provide the response to a question if the application domain is unknown i.e. **U**nknown **A**pplication **D**omain (UAD). The attempt to find a response for UAD has been undertaken already by Google [11]. The goal of **the Google SMS service** is to provide the large existing base of users with access to the types of information they are most likely to need when mobile. Users simply send their query as a text message and receive results in the reply. Simple conventions are used to express queries for phone book listings, dictionary definitions, product prices, etc. Google touts this service as ***"Just text. No links. No web pages"*** [10]. Simply the answers one is looking to find. Examples of queries are *"pizza 21228"* to find pizza places located near to the University of Maryland, Baltimore County (UMBC), or *"george bush, washington dc"* to find the address and phone number of the US president or *"Price ipod 20gb"* to get a list of prices (and sellers) for an ipod. The **M**obile **Q**uery (MQ) is sent to the 5-digit US short code 46645, which corresponds to Google on most phones. One could, of course, leverage existing mobile technology through the WAP browser on mobile phones as an alternative to using SMS. For example, if you want to find out the dictionary definition of the word *"spring"* simply enter message ***"define:spring"*** to search in your mobile "Google" on WAP enabled phones. The following response will be displayed: *"the season of growth; "the emerging buds were a sure sign of spring"; "he will hold office until the spring of next year""*. If the same query is repeated but on a personal computer rather more information regarding the definition will be displayed (the most compact part of those information is shown below).

Definitions of **spring** on the Web:

- the season of growth; "the emerging buds were a sure sign of spring"; "he will hold office until the spring of next year";
- a natural flow of ground water;
- jump: move forward by leaps and bounds; "The horse bounded across the meadow"; "The child leapt across the puddle"; "Can you jump over the fence?";
- form: develop into a distinctive entity; "our plans began to take shape";
- a metal elastic device that returns to its shape or position when pushed or pulled or pressed; "the spring was broken";
- leap: a light, self-propelled movement upwards or forwards;
- bounce: spring back; spring away from an impact; "The rubber ball bounced"; "These particles do not resile but they unite after they collide";
- give: the elasticity of something that can be stretched and returns to its original length;
- develop suddenly; "The tire sprang a leak";
- a point at which water issues forth;
- produce or disclose suddenly or unexpectedly; "He sprang these news on me just as I was leaving".

Despite the quite impressive results using MQ there remain some significant problems:

- **Quality not quantity**. The best conformity between the returned response and the MQ is more important than the quantity of information found. How can we define the **quality of response**? When a user asks for the definition ***of spring*** Google sent the first line of those definitions to the user's mobile. Is it what the user anticipated, or would it be more important for him to know that *"spring is a metal elastic device…"*? Using a single SMS restricts the answer to 160 characters, and depending upon the type of phone used the display area is very likely to be much less than this. Although *"no answer is worse than a bad answer"*. A disappointing response could deter the customer from using the SMS service again.

- **No dialogue**. Let us distinguish between QAS and Mobile QAS (MQAS). The principal difference is that QAS allows for a clarification dialogue [3,4] with the user to provide the best possible response through interaction but MQAS does not. As for MQAS only a single short user MQ must be used to create the response.
- **Ungrammatical MQ.** The dream about MQ: *"Be precise and informative about your problem. Write in clear, grammatical, correctly-spelt language"* – is far from being realised. As a rule an MQ will be ungrammatical, not because users are illiterate but because most users are lazy i.e. they want to achieve the desired result by using the minimum effort. For example, they do not want to use upper case to type MQ *"george bush, washington dc"* [9] or use dots to separate *"d"* and *"c"*. Other examples of typical grammatical errors are: typing *"its"* instead of *"it's"*, *"lose"* instead of *"loose"*, or *"discrete"* instead of *"discreet"*. The fact is that ***MQ simply cannot be spelt, punctuated, and capitalised correctly*** but the main requirement for MQAS is - to handle non-standard or poorly formed/structured (but, nevertheless, **meaningful**) user's MQ.
- **Fuzzy MQ.** The users are generally unable to describe completely and unambiguously what it is they are looking for [4]. For example, the MQ *"Who won the 2006 World Cup?"* is absolutely clear from the user's point of view because it goes without saying that *football* is implicit in the question. Fuzzy MQ requires **fuzzy searching**. Fuzzy response searching is a technique for finding a reply that approximately matches the search MQ.
- **Facts Ambiguity.** The problems of answering the MQ depends not only on an incomplete and/or ambiguous MQ but also on *facts ambiguity*, especially when MQAS, as a result of searching information on the internet, expands its local KB by adding appropriate facts. It is not easy to recognise ***semantic***, ***lexical*** or ***structural*** ambiguity of facts. For example, the fact *"Stolen painting found by tree"* has structural ambiguity and can be interpreted as: (1) *A tree found a stolen painting*; (2) *A person found a stolen painting near a tree*. The MQ *"Who stole a painting?"* requires some effort from MQAS to reply correctly (we will discuss this in the *Mobile Queries Classification* session).

The meaning of an MQ depends not only on the things it describes, explicitly and implicitly, but also on both aspects of its causality: what caused it to be said and what result is intended by saying it. In other words, the meaning of an MQ depends not only on the MQ itself, but on **who** (age, sex, nationality) sends it and **when**, **where**, **why** and to **whom** it is sent. Such information helps to understand ***what*** the user meant in his/her MQ. In this paper we shall discuss **why** the user sent the MQ and the determination of some of the constraints the MQ imposes on a possible answer.

Why was the Mobile Query Sent?

Why do people not ask questions? It is very important to know why, especially if you want to learn and improve performance in an organisation. There are, as minimum, some 10 reasons why people never ask questions [12]. But here we want to find out the answer to the opposite question: *why do people ask questions?* While it is possible to question without thinking, it is impossible to think without questioning. A good question might just provide the means for overcoming a particular obstacle and achieving a stated goal or objective. There are some 12 reasons identified for asking questions [13]. This list is based on three assumptions: (1) a genuine question is one to which the answer is not known; (2) curiosity is the driver; and (3) the questions are intended in a constructive manner, i.e. to intimidate, dominate, demonstrate how smart one is, or to prove that one is right. These reasons are:

1. Gather information.
2. Build and maintain relationships, participate in communication.
3. Learn, teach, and reflect.
4. Think clearly, critically, and strategically.
5. Challenge assumptions.
6. Solve problems and make decisions.

7. Clarify and confirm information heard.
8. Negotiate and resolve conflicts.
9. Set and accomplish goals.
10. Take charge and focus attention on yourself.
11. Create and innovate – open new possibilities, provoke thought in others.
12. Catalyse productive and accountable thinking, conversation, and action.

The reason behind asking questions depends on the situation when the question has been asked. Let us enumerate some obvious situations:

Ask the audience a verbal question, e.g. a teacher questioning pupils.

Ask a person a verbal question but in front of an audience, e.g. at a scientific conference.

Ask a person a written question but in front of an audience, e.g. at a scientific conference.

Ask a person an anonymous written question but in front of an audience, e.g. at a company's meeting.

Ask a "face to face" question.

Ask a "face to face" question on the phone.

Ask an audience a written question, e.g. through the internet.

Ask a personal written question, e.g. through email.

Ask an SMS question. It is completely different in comparison to the previous point because of size, convenience and money.

Ask the Artificial Intelligent Question-Answering System (AIQAS) a verbal question.

Ask the AIQAS a MQ. This is exactly our case and just **two** reasons (1 and 6) among the 12 listed earlier are appropriate for this particular situation.

On the one hand two reasons to send an MQ have been selected, on the other hand there should be some external, implicit reason that causes the user to send an MQ. Let us call such a reason a **meta-reason**.

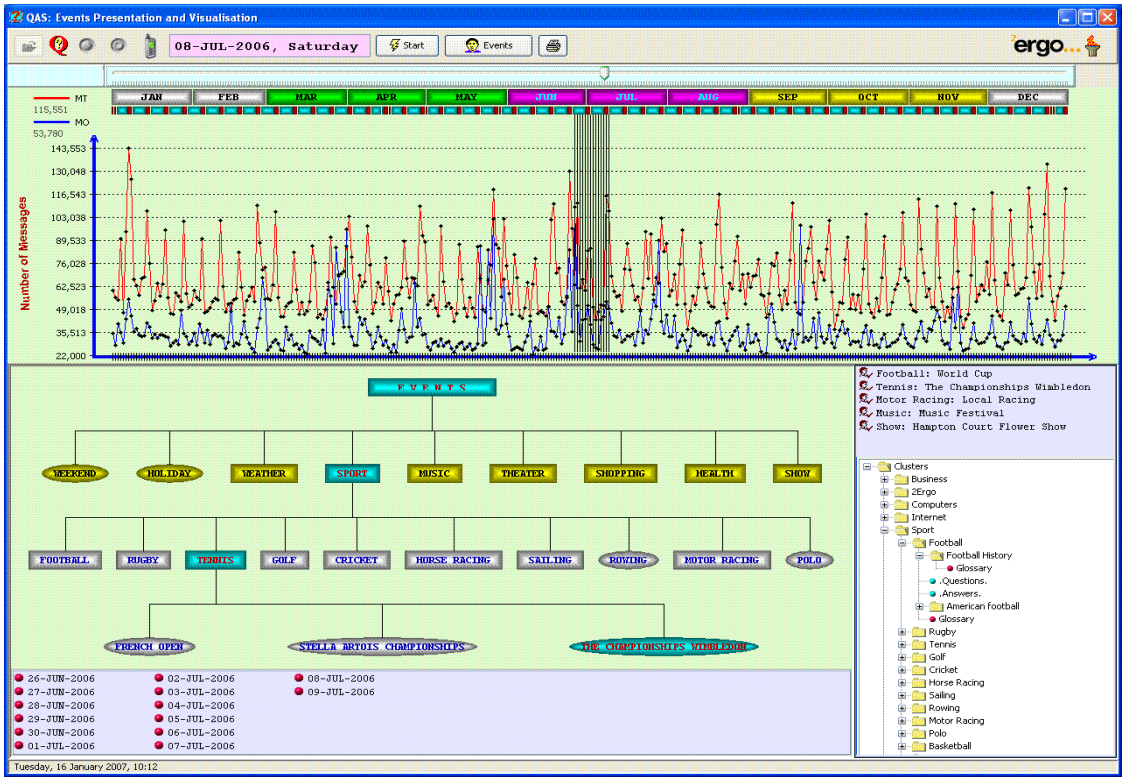


Figure 1. SMS-activities and Events

Meta-reason for Mobile Query Sending

It is very important to underline that in this paper we consider the precise situation when an MQ is sent to the AIQAS but not to another person. Initially, we decided to find out about the SMS-activities during the year i.e. it to discover whether and how SMS-activities depend on seasons, weekdays, holidays, events etc. (see Figure 1). Red (top) charts represent quantity of daily SMS. The set of events is represented as **events hierarchical structure**, nodes of which, in essence, are clusters and have links to corresponding nodes of **clusters structure**. Suppose an MQ was sent on Saturday, 08.07.06. MQAS will extract a list of events taking place on or around this day from the KB (see Figure 1) and create a list of potentially active clusters: for example, ***football, tennis, motor racing, music,*** and ***show***. Thus, MQAS has selected five WDAD ahead of the MQ analysis. Of course, for MQ: *"What tablets can help me with headache?"* these WDAD would not be used but they might help to disambiguate MQ related to one of these enumerated clusters.

Mobile Queries Classification

As part of its MQ processing component, the MQAS first attempt to classify each MQ by their type and represented as a pair: **<Question Type>** ↦ **<Answer Type>** [15]**.** Only two kinds of question are allowed for MQ: (1) **Specific question** (*what, who, why, how, where, when*, etc.) and (2) **Yes-No question.** For example, instead of asking: *"Name all directors of 2ergo"*, the specific question *Who* should be used: *"Who are directors of 2ergo?"*. Examples of MQ classification are shown below:

- What ↦ (money, definition, name, etc): "What was the monetary value of the Nobel Peace Price in 1989?"
- What ↦ (person, organisation): "What costume designer decided that M. Jackson should wear only one glove?"
- What ↦ (date): "In what year did Ireland elect its first woman president?"
- What ↦ (location): "What is the capital of the Ukraine?"
- Which ↦ (person, organisation): "Which former Klu Klux Klan member won an elected office in the U.S.?"
- Which ↦ (date): "In which year was New Zealand excluded from the ANZUS alliance?"
- Which ↦ (location): "Which city has the oldest relationship as sister-city with Los Angeles?"
- Who ↦ (person, organisation): "Who is the author of the book "The Iron Lady: A Biography of Margaret Thatcher"?"
- Whom ↦ (person, organisation): "Whom did Italy beat in the final of 2006 Football World Cup?"
- Where ↦ (location): "Where is Kharkov?"
- When ↦ (date): "When did the Jurassic Period end?"
- Why ↦ (reason): "Why do people shake hands to show friendliness?"
- How ↦ (action): "How did Socrates die?"
- How many ↦ (number): "How many James Bond novels are there?"
- How much ↦ (money, price, time, etc.): "How much pizza do Americans eat in a day?"
- How long ↦ (time, distance): "How long does it take to travel from Plymouth to Rawtenstall?"

As for a Yes-No MQ, in general, it is better to avoid asking a Yes-No MQ unless one wants to receive a yes or no answers. For example, instead of asking *"Are there any tablets to relieve my headache?"* it is better to ask *"What tablets can relieve my headache?"*. A Yes-No MQ highlights a very important problem of communication with both

a “human” QAS or with AIQAS. By default, when a user sends an MQ he/she does not doubt that the recipient is a human being. That is why the Yes-No MQ like “*Does anyone know ... ?”* , “*Can someone ... ?*“, “*Is it possible to ... ?”, etc* is very popular. In a call centre the recipients appear to find such MQ annoying [16] — and are likely to return logically impeccable but dismissive answers like *“Yes, you can be helped”* in response to the MQ *“Can you help me to find an address of 2ergo?”*. As for our MQAS, on the one hand, we decided to give the impression that our system is not artificial, on the other hand, MQAS is always “well behaved” (i.e. it is simply impossible to annoy it – since it is a machine) and that is why MQAS would reply to user *“Yes, you can be helped but it is better to ask a specific question”*.

The MQ classifier is critical during the stage of an MQ’s type recognition (see Figure 2). A list of supplementary key words for each *question word* allows the specification of the type of expected response more precisely (see an example of ***When*** description in **Knowledge Base Structure** session).

Derivation of implicit information for MQ Response

The primary focus of MQAS is for the creation of a coherent, understandable answer that is responsive to the originally posed MQ. The factually explicit MQ e.g. *“What is the Taj Mahal?”* does not require a great deal of effort to create the answer if the MQAS KB contains the fact: *“Taj Mahal is a beautiful mausoleum at Agra built by the Mogul emperor Shah Jahan (completed in 1649) in memory of his favourite wife”*. But the problems are, firstly, it is impossible beforehand to classify MQ as **explicit** or **implicit**, and secondly, in reality most of the MQs are implicit. For example, if the MQAS KB contains the fact: *“Aleksandr Sergeyevich Pushkin is a Russian poet (1799 -1837)”*. The obvious implicit MQs related to this fact are *“When was Pushkin born?”* or *“When did Pushkin die?”*. Below we discuss the possible ways for the derivation of implicit information from the KB.

In this paper we continue to apply a ***psycholinguistic approach*** to natural language (NL) processing [1]. The only system truly capable of adequate understanding of an MQ is human. What is more, children seem to use NL effortlessly in spite of *not knowing the grammar*. Parsing was taught in school as an algorithmic task. For derivation of implicit information a ***natural inference engine*** based on *human reasoning* is used. *Human reasoning* might be described using fuzzy attributes: *approximate, common sense, default, enumerative, evidential, hypothetical, inexact, integrating, plausible, procedural, taxonomic.* In other words, instead of logically making some conclusion a human would very often ***justify*** the decision based on maximum argumentation about problem solving within his/her knowledge. At present it is understandable that the real knowledge base (KB) is ***incomplete***, ***inconsistent, inaccurate*** and ***open***. These facts do not necessarily permit the use of a traditional logical approach. For example, assume that the following two statements are in the KB: (1) *“A student likes to read a detective story”,* (2) *“A student does not like to read mathematical books”* and the MQ simply states *“Does a student like to read?”*.

The derivation of implicit information is not always based on the logical operations:

- From the conditional judgement *"If '**A**' then '**B**'"* it is not possible to draw the conclusion *"If not '**A**' then not '**B**'"*. As for the sentence, *"If you smoke you will fall ill"* the conclusion *"If you do not smoke you will not fall ill"* is not true since non-smokers also become ill. Representation of the conditional judgements stated in the linguistic form and the conclusions drawn from them depend widely on their contents.
- The examinees were given the sentences: *"Acorns always grow on an oak"* and *"Acorns grow on a tree"*, they concluded on this basis: *"This tree is an oak"* [14]. The potential fallacy of this conclusion becomes evident when considering another example apparently having the same structure. From the sentences, *"All football-players are good runners"* and *"Peter is a good runner,"* one can hardly draw the conclusion that *"Peter is a football-player"*. In the first instance, at first glance, the rule of inadmissibility of the general use judgement was violated. It is not possible to conclude that *"All **B**'s are **A**'s"* from the judgement *"All **A**'s are **B**'s"*. It should be specified that the simple transformation of the general judgement is not always incorrect but only in those cases when the participants of the judgement have dissimilar size. In the case of the identical sizes the simple transformation of a general judgement is possible. For instance, the

judgement *"All squares are equilateral rectangles"* transforms into the judgement *"All equilateral rectangles are squares"* as the "participants": *"square"* and *"equilateral rectangle"* are of equal size. Really, there are no rectangles other than equilateral ones being squares. As to the judgement *"All football-players are good runners"* this is not the case. Here the "participants" sizes: *"football-players"* and *"good runners"* are different and the simple transformation of this judgement is inadmissible.

Cognitive transformations as the natural inference source for the derivation of implicit information:

- Taking into account the cognitive transformations based on the semantic relations featuring the properties of **symmetry**, **transitivity** and **additivity** allows MQAS to answer a Yes-No type MQ. For example:

  (1) ***Fact:*** *"'**A**' is a colleague of '**B**'".* ***MQ:*** *"Is '**B**' a colleague of '**A**'?".*

  (2) ***Fact:*** *"'**A**' buys a doll".* ***MQ:*** *"Does '**A**' buy a toy?".*

  (3) ***Fact:*** *"'**A**' is a teacher of '**B**'".* ***MQ:*** *"Is '**B**' a pupil of '**A**'?".*

- It is important to process facts on the things belonging to transformation. For example, from the fact: *"Mary has given a book to John after New Year"* three different MQ might be answered:

  (1) *"Did John have this book before New Year?";*

  (2) *"Does John has this book now?";*

  (3) *"Does Mary have it now?".*

The derivation of implicit information based on *human reasoning* is a part of Reply Search Engine (RSE). The central question to be addressed by any QAS is how the storage of information is organised in KB and we now turn to consider this.

Knowledge Base Structure

In the general case, under the *knowledge base structure* one should understand the regularity of data distribution in memory assuring the storage of various links between separate elements of stored information. At every moment KB deals only with relatively *small fragments* of the external world. So, the corresponding structures are needed to integrate these fragments separated in time into the integral picture. The structures obtained as a result of integration should contain more information than had been used for its creation. The organisation of knowledge storage should make allowance for such features of the human memory as [17]:

- associativity;
- ability to reflect similar features for different objects and different features for similar objects;
- hierarchical and heterarchical organisation of information. The idea of heterarchical approach correlates naturally with the human ability to use all kinds of information in the process of natural language understanding. As Quillain remarks "*... a full association structure... forms simply large, very complicated net of nodes and unidirectional memory links between them... The predetermined hierarchy of "super-" and "subclasses" is absent; every word is a "patriarch" in its own hierarchy if some process of search initiates with it. Analogously every word is in different places within hierarchies of large diversity of wordy concepts if the search process starts with them*" [18, p.5];
- associative relations weight variable;
- representation of the environment statistical properties;
- independence of the knowledge extraction time from the volume of knowledge being stored in this memory;
- the knowledge cannot realistically be regarded as a static resource, to be accumulated and stored within a system. It is a formative, *self-organising* character, with the ability to change the organisation within which it is held.

The attempt to take into account these features of human memory in MQAS KB has been undertaken. The structure of KB is shown in Figure 2. MQAS's consists of six different structures:

- **Direct** and **Reverse Dictionary** (DD and RD respectively) are represented by an L-Tree structure. DD is an initial structure which provides the recognition of *new* words, the normalisation of *known* words and the direct links with the corresponding nodes of **Semantic** structure and **Factual Knowledge** structure. RD together with DD is used to correct wrong words, spelling them automatically.
- **Semantics** structure (or Sequential-Simultaneous Structure (SSS)) provides the *sequential* and *simultaneous* analysis of string information and handling of *new* and/or *known* sentences or combinations of words. SSS is a special combination of hierarchical and network structures. Each of the elements in SSS are associated with several other elements logically including itself or included by it. With the successive presentation of facts as a sequence of words the strongest relation in it is the relation between the nearest neighbouring words. *"Their succeeding one after another presents evidently an important condition of structuring"* [19, p.231]. The ***sequential*** part of SSS provides ***hierarchical*** organisation of information and the ***simultaneous*** - ***heterarchical*** organisation of information

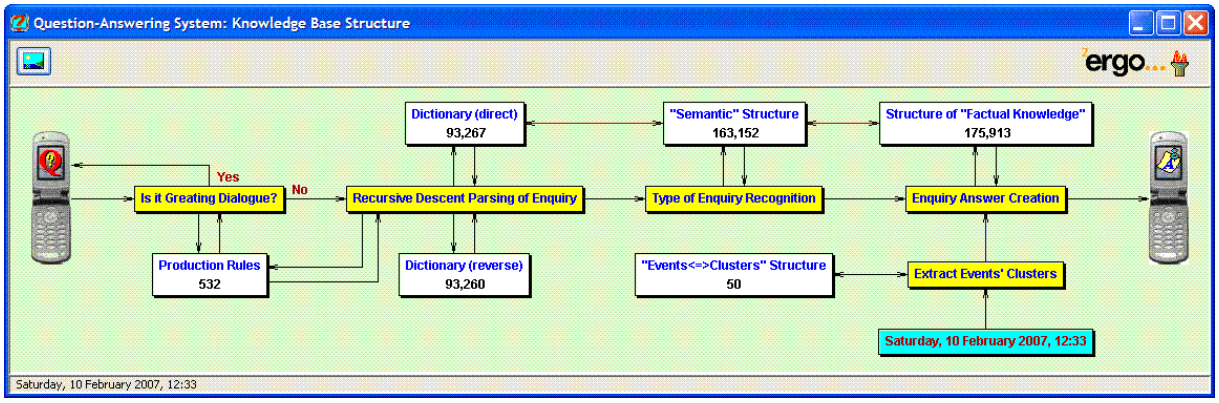


Figure 2. Knowledge Base Structure

- **Events-Clusters** Structure (ECS)
- **Factual Knowledge** Structure (FKS)
- **Production Rules** Structure (PRS) [4]. New class of PR **Question** has been added. The example of PR in format: **<Class>** $\mapsto$ **<Antecedent>** $\Rightarrow$ **<Consequent>** is shown below.

Question $\mapsto$ **When** $\Rightarrow$ **0:**born,start,begin,commence,come,become;**1:**live,interval,period;

2:die,end,finish,stop;**3:**occur,happen,find.

Let us come back to a fact: *"Aleksandr Sergeyevich Pushkin is a Russian poet (1799 -1837)"* in FKS. Assume the MQ *"When was Pushkin born?"* has been asked. In the consequent of PR ***Question*** $\mapsto$ ***When*** subclass **0** has been found, which means that the left-hand number from the interval should be selected as a birth year. First of all MQAS tries to find in FKS the direct fact for an answer like *"Pushkin was born in 1799"*. If such a fact does not exist MQAS starts to analyse facts with date's interval and then from the appropriate fact select left number as a birth year i.e. *"In 1799"*.It is worth noting that, MQAS always try to minimise the length of the response because of the display constraints of the mobile device and the SMS message.

At the end of this session it is important to emphasise:

- Local KB of any MQAS cannot be **complete**, in other words, we shall never be able to establish **information completeness** of KB.
- MQAS should be **self-learning** i.e. when MQAS cannot find a suitable response in the local KB it should search using the **Internet** and then not only send the reply to the questioner but also **automatically**

extend the local KB. The searching of information across the Internet represents the greatest problem because the result may identify a huge set of documents among which the appropriate response will need to be found, We shall discuss this problem in the next paper.

Mobile Query Parsing

The main requirement for MQAS is to reply to any (even non-standard or poorly formed but, nevertheless, **meaningful**) user's MQ. But it is not easy to find the answer even for an ideal MQ because for an artificial intelligent system like MQAS the power of natural language to describe the same events but in quite different ways is a great problem. For example, the primitive action: *"take by theft"* might be described as: *"hook", "snitch", "thieve", "cop", "knock off", or "glom"*. The main purpose of MQ processing is to understand ***what was meant*** rather than ***what was said***. The mechanism of query parsing is very simple: *"eliminating the unnecessary until only the necessary remain"* and has been discussed elsewhere [2]. Here we just remind ourselves of the main steps involved in MQ processing. This MQAS takes the MQ as a character sequence, locates the MQ boundaries, and converts the original MQ to a *skeleton*. Such conversion will require several steps:

- MQ related synonyms' dictionary creation to equate synonymous words and phrases, such as "NLP" and "Natural Language Processing";
- Irregular verb normalisation. Once the word has been identified then it should be changed back to its simplest form for efficient word recognition. For example*, writes, writing, wrote, written* will be changed to *write* and the corresponding attributes of the original form will be saved;
- Noisy (non-searchable) word elimination;
- Plural to singular conversion.

The skeleton of the MQ is matched against all relevant data in SSS to find the appropriate links to FKS. The result of such searching is shown on Figure 3.

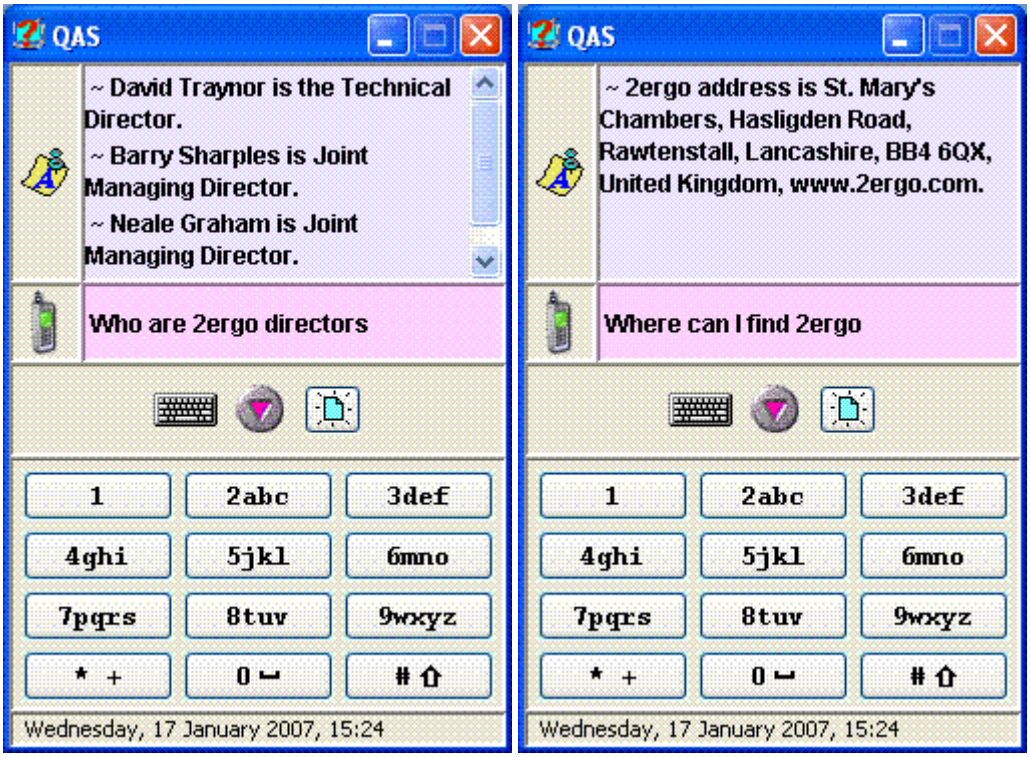


Figure 3. The examples of MQAS responses

Conclusion

The possible solutions of how to provide the response to a question if the application domain is unknown have been considered. In the result of that, the MQAS effectively places information directly into the hands of any users - eliminating the need for technical support specialists continually to address *ad hoc* requests from end users. The MQAS are addressed from both scientific and industrial perspectives. Whether MQAS is searching the local KB, or the worldwide web, MQAS understands the relationships between words, enabling it to extract all key concepts and automatically build a semantic index organised in a problem-solution format. Because MQAS extracts and organises the content, the user receives specific and relevant answers to his/her MQ — not a list of documents.

Bibliography

[1] G.Coles, T.Coles, V.A.Lovitskii, "Natural Interface Language", Proc. of the VIII-th International Conference on Knowledge-Dialogue-Solution: KDS-99, Kacivelli (Ukraine), 104 -109, 1999.

[2] T.Coles, V.A.Lovitskii, "Text Searching and Mining", J. of Artificial Intelligence, National Academy of Sciences of Ukraine, Vol 3, 488-496, 2000.

[3] D.Burns, R.Fallon, P.Lewis, V.Lovitskii, S.Owen, "Verbal Dialogue Versus Written Dialogue", International Journal "Information Theories and Application", Vol 12(4), 369-377, 2005.

[4] G.Francis, M.Lishman, V.Lovitskii, D.Traynor, "Instantaneous Database Access", Proc. of the XII-th International Conference on Knowledge-Dialogue-Solution: KDS-2006, Varna (Bulgaria), 314 - 322, 2006.

[5] K.Braithwaite, M.Lishman, V.Lovitskii, D.Traynor, "Distinctive Features of Mobile Messages Processing", Proc. of the XII-th International Conference on Knowledge-Dialogue-Solution: KDS-2006, Varna (Bulgaria),322 - 329, 2006.

[6] www.hsbc.com.sg/1/2/personal/services/sms-from-hsbc

[7] www.2ergo.com (2safeguard™, 2NC™)

[8] www.revenue.ie

[9] www.smseducation.org

[10] www.edinburghairport.com

[11] Google SMS query service

[12] www.Think6Results.com

[13] M.Adams, Inquiry Institute, www.inquiryinstitute.com

[14] R.J.Harris, G.E.Monaco, "Psychology of Pragmatic Implication: Information Processing between the Lines", J. Exp. Psychol. General, 107, 1-22, 1978.

[15] D.Moldovan, S.Harabagiu, et al. "Lasso: A Tool for Surfing the Answer Net.", TREC-8 Draft Proceedings, NIST, 65-73, 1999.

[16] www.82ask.com.

[17] V.A.Lovitskii and K.Wittamore, "AMONIL: Achievement of Multiple Objectives using a Natural Interface Language", Proc. of the International Joint Conference on Knowledge-Dialogue-Solution: KDS-97, Yalta (Ukraine), 270-281, 1997.

[18] M.R.Quillian, "Word concepts: A theory and simulation of some basic semantic capabilities", C.I.P. working paper 79, Carnegie Inst. of Technology, Pittsburgh, 1965.

[19] J.Hoffmann, "Das Aktive Gedachtnis. Psycologische Experimente und Theorien zur Menschlichen Gedachtnistatigkeit", VEB Deutscher Vergal der Wissenschaften, Berlin, 1982.

Authors' Information

Vladimir Lovitskii *– 2 Ergo Ltd, St. Mary's Chambers, Haslingden Road, Rawtenstall, Lancashire, BB4 6QX, UK, vladimir@2ergo.com*

Michael Thrasher *– University of Plymouth, Plymouth, Devon, PL4 6DX, UK, mthrasher@plymouth.ac.uk*

David Traynor *– 2 Ergo Ltd, St. Mary's Chambers, Haslingden Road, Rawtenstall, Lancashire, BB4 6QX, UK, david.traynor@2ergo.com*

ANALYSIS OF TEXT DOCUMENTS IN AUTOMATIC ABSTRACTING SYSTEM

Stanislav Lipnitsky, Denis Nasuro

***Abstract**: Mathematical model of syntactic and semantic analysis of text documents is offered. On the base of this model a procedure of the detection of informative sentences in text documents is realized. Results of modeling are used in the computer system of automatic abstracting of big text documents.*

***Keywords**: formal grammar, syntagma, syntactic tree, text analysis, corpus of texts, informativity, text abstracting.*

***ACM Classification Keywords**: I.2.7 Natural Language Processing - Text analysis*

Introduction

The main purpose of the automatic abstracting system is the intellectual analysis of big text documents. The syntactic and semantic analysis of the text goes first regardless of analytical processing methods using in the system. It consists in building of syntactic structure and finding of semantic characteristic for each sentence.

Method of syntactic and semantic analysis of the text is described in the article. This method is based on modeling of syntagma detection in the text by means of special formal grammar and knowledge base of object domain. Knowledge base is represented as a situation-syntagmatic network that consists of informative syntagmatic structures and their situational bindings.

Text syntactic analysis

Stroke-grammar.

Let $F = \langle V, N, I, R \rangle$ – a formal generative grammar, where V – *a* nonempty set of terminal symbols (will name them *words*), $N = \{I, '\}$ – a set of nonterminal symbols, I – a start symbol, and R – a grammar schema, i. e. a set of derivation rules $\alpha \to \beta$ (α and β – different chains in the dictionary $V \cup N$). Schema R of grammar F is defined as follows:

1) For any word $a \in V$ there are derivation rules $I \to a'$ and $a' \to a$;

2) Remaining derivation rules are of the form $a' \to a'b'$ or a' $\to b'a'$, where $a, b \in V$.

Symbol «'» (stroke) is included in nonterminal symbols for convenience. That's why grammar F is named a *stroke-grammar* with dictionary V [Кравцов, 2005]. Generated by stroke-grammar language $L(F)$ is named a *source language*, chains of this language will be called *source language sentences* or *source sentences*. Dictionary V will be called a *source language dictionary* or a *source dictionary*. Any nonempty fully ordered subset of the language $L(F)$ will be defined as a *text* of this language or a *source text*.

Syntagms and syntagmatic structures.

We shall use a relation of syntactic subordination when modeling a syntactic structure of sentences of the language $L(F)$. This relation will be defined as follows.

Let $\pi = a_1 a_2 ... a_n$ – is a arbitrary sentence of the language $L(F)$, where $a_1, a_2, ..., a_n$ – are words of the sentence. Some nonempty non-overlapping (that have no common words) subchains of the sentence π are denote by μ and ν. We shall name a binary relation Ω_π in a set of all such subchains of the sentence π as *a relation of syntactic subordination* in the sentence π of the language $L(F)$ if:

1) For any words a_i, a_j $(i, j = \overline{1, n};\ i \neq j)$ of the sentence π $(a_i, a_j) \in \Omega_\pi$ if and only if there are subchains $\alpha a_i' \beta$, $\gamma a_i' a_j' \delta$ (or $\gamma a_j' a_i' \delta$), and $\alpha a_i' \beta \Rightarrow^* \gamma a_i' a_j' \delta$ at that (or $\alpha a_i' \beta \Rightarrow^* \gamma a_j' a_i' \delta$) in derivation of sentence π from start symbol I. Here $\Rightarrow^*$ is a symbol of derivability in the grammar F, and $\alpha, \beta, \gamma, \delta$ are chains of the dictionary $V \cup N$.

Some of chains α, β, γ, δ could be empty (possibly all). If $i < j$ (or $j < i$), than chain a_ia_j (or a_ja_i) will be called *a syntagma* of the sentence π of the language $L(F)$. When $j \neq i+1$ (or $i \neq j + 1$) syntagma a_ia_j (or a_ja_i) will be called *separated*, and when $j = i + 1$ (or $i = j + 1$) – *unseparated*;

2) For arbitrary nonempty non-overlapping subchains μ and ν of the sentence π $(\mu, \nu) \in \Omega_\pi$ if and only if there is such syntagma a_ia_j of the sentence π, that in derivation of the sentence π from the start symbol I the chain received from a_i', and the chain ν – from a_j'. Let us denote by $\prec$ full order in the set of all nonempty non-overlapping subchains of the sentence π, corresponding to words natural order, i.e. that for all $i, j = \overline{1, n-1}$, $r, s = \overline{1, n}$ $a_ia_{i+1}... a_j \prec a_ra_{r+1} ... a_s$ if and only if $j < r$. If $\mu \prec \nu$ (or $\nu \prec \mu$), then the subchain $\mu\nu$ (or $\nu\mu$) we shall call *syntagmatic stucture* of the sentence π of the language $L(F)$. In this case let us say that μ – *determined*, and ν – *determining* members of syntagmatic structures $\mu\nu$ and $\nu\mu$.

Union $\Omega_{L(F)} = \bigcup_{\pi \in L(F)} \Omega_\pi$ of relations of syntactical subordination in all sentences of the language we shall call *relations of syntactical subordination* in the language $L(F)$. Syntagms and syntagmatic structures of sentences of this language we shall call syntagms and syntagmatic structures of the language $L(F)$.

Syntactic tree of sentence.

If ab – is a syntagma of a certain sentence of the language $L(F)$ and $(a, b) \in \Omega_\pi$, then let us say that syntactical binding is *directed* from word a to word b. If $(b, a) \in \Omega_\pi$, then such binding is oppositely directed. Let us denote direction of syntactical binding of words by arrow that starts over determined member of the syntagma and ends over determining syntagma member (for instance, $\alpha\overrightarrow{a\beta b}\gamma$, $\alpha\overleftarrow{a\beta b}\gamma$). If direction of syntactical binding is unknown or insignificant then we shall denote it by the line over a syntagma (for instance, $\alpha\overline{a\beta b}\gamma$).

Usually syntactical bindings of words in the sentence are represented as a directed graph, nodes of graph are words, links correspond with syntactical bindings. We shall define formal notion of syntactical graph as follows.

A directed graph of relation Ω_π on the set of all words of the sentence π we shall call a syntactical graph of the sentence π. A syntactical graph of the sentence that includes only one word a we shall consider graph $(\{a\}, \varnothing)$. A syntactical graph of any chain δ, derived from the sentence π by transposition of words in it we shall call syntactical graph of the sentence π.

Let us see what form has a syntactical graph of the sentence that belongs to the source language $L(F)$.

The statement 1. *A syntactical graph of any sentence of the language $L(F)$ is a directed tree* (let call it *a syntactical tree*).

The proof. Let us prove it by method of mathematical induction. If $n = 1$ and $n = 2$ then syntactical graphs of word and syntagma are directed trees. Let us assume that if $n = k$ then syntactical graph of the sentence with k words is a directed tree. Let us prove that after adding one more word to the sentence, i.e. if $n = k + 1$, then syntactical graph of the sentence is still a directed tree. We shall denote added word by b. Then according to the definition of grammar F there is a word a in the sentence that is an determined member of syntagma ab or ba. If knot a of a directed tree with k knots connect with knot b by link (a, b) then it's obviously that we have a directed tree again. Q.E.D. The statement 1 is proved.

Marginal syntagms.

Let $\alpha a\beta b\gamma$ (or $\alpha b\beta a\gamma$) – is a arbitrary sentence of the language $L(F)$, where α, β, $\gamma \in V^*$ (V^* – is a set of all chains in the dictionary V of the grammar G), ab (or ba) – is a syntagma of this sentence with a determined member a and determining b.

Let us call syntagma ab (or ba) a *marginal syntagma* of the sentence $\alpha a\beta b\gamma$ (or $\alpha b\beta a\gamma$), if chains bc and cb are not syntagms for any word c ($c \neq b$) of the sentence [Липницкий, 2005]. Word b of the syntagma ab or ba we shall name a *marginal word* of syntagms αb and ba.

Properties of marginal syntagms.

Let δ – is a arbitrary chain of set V^+ of all nonempty chains in dictionary V, and certain sentence π of language $L(F)$ is its subchain. If $\mu\nu$ – is a syntagmatic structure of the sentence π, then we shall consider it also as a *syntagmatic structure of chain* δ. If *ab* (or *ba*) – is a marginal syntagma of sentence π with marginal word *b*, such that for any word *c* ($c \neq b$) of chain δ pairs *bc* and *cb* are not syntagms, then *ab* (or *ba*) we shall name a *marginal syntagma of chain* δ.

Lemma. *If* $\rho \in V^+$, *and ab* (*or ba*) *– the marginal syntagma of chain* ρ, *and in schema R of grammar F there is a derivation rule* $a' \to a'b'$ (*or* $a' \to b'a'$), *then chain* σ *received from* ρ *by removal determining member b of syntagma ab* (*or ba*) *is the sentence of language L*(*F*) *if and only if* $\rho \in L(F)$.

The proof. *Necessity.* Let chain σ is the sentence of the language $L(F)$. Then necessity, i.e. existence of relation $\rho \in L(F)$ follows from the fact of existence in schema *R* of grammar *F* of derivation rules $a' \to a'b'$ (or $a' \to b'a'$) and $a' \to a$, $b' \to b$.

Sufficiency. Let there is a syntagma *ab* with a determined member *a* and determinating member *b*. Then for chain ρ there is a derivation $W = (I, \alpha, \beta, \dots, \gamma, \mu a'\nu, \mu a'b'\nu, \dots, \mu ab\nu, \dots, \rho)$ in grammar *F* where $\alpha, \beta, \gamma, \mu, \nu \in V^*$. As *ab* – is a marginal syntagma of sentence ρ then, as follows from marginal syntagma definition, for any word *c* of sentence ρ chain *bc* is not a syntagma, i.e. when derive a sentence ρ rules like $b' \to b'c'$ are not used and chain $\mu a'b'\nu$ in derivation *W* is received from chain $\mu a'\nu$ by applying derivation rule $a' \to a'b'$. If we shall exclude a chain $\mu a'b'\nu$ from the derivation *W* then we shall recieve a derivation of chain σ from start symbol *I*. The case when chain *ba* is a syntagma of sentence ρ could be considered similarly. The lemma is proved.

Using this lemma it is easy to prove the following

The statement 2. *If* $\mu a'b'\nu$ (*or* $\mu b'a'\nu$) *– some chain in dictionary V*, *where* $\mu, \nu \in V^*$, *ab* (*or ba*) *– a marginal syntagma with determining member b*, *and in schema R of grammar F there is derivation rule* $a' \to a'b'$ (*or* $a' \to b'a'$), *then chain* $\mu a'\nu$ *could be raised to start symbol I of grammar F if and only if chain* $\mu a'b'\nu$ (*or* $\mu b'a'\nu$) *could be raised to symbol I*.

The proof. *Necessity.* Let chain $\mu a'\nu$ could be raised to start symbol *I*. Let us prove that chain $\mu a'b'\nu$ could be raised to symbol *I*. Indeed, applying a derivation rule $a' \to a$ to chain $\mu a'\nu$ we shall get a chain $\mu a\nu$, that is a sentence of language $L(F)$, this implies chain $\mu a'b'\nu$ could be raised to symbol *I* because there are derivation rules $a' \to a$, $b' \to b$. Necessity of chain $\mu b'a'\nu$ could be proved the same way.

Sufficiency. Let now chain $\mu a'b'\nu$ could be raised to start symbol *I*. The proof that chain $\mu a'\nu$ could be also raised to this symbol follows from sufficiency of lemma. Let us apply derivation rules $a' \to a$, $b' \to b$ to chain $\mu a'b'\nu$. We shall get a sentence $\mu ab\nu$ of language $L(F)$. By virtue of lemma, chain $\mu a\nu$ is also a sentence of the language, this implies chain $\mu a'\nu$ could be raised to symbol *I*. The proof that chain $\mu a'\nu$ could be raised to symbol *I* (if chain $\mu b'a'\nu$ could be raised to this symbol) is similarly. The statement 2 is proved.

According to the statement 2 the algorithm of source chain syntactical analysis could be constructed in the form of cyclic reduction process to start symbol by a principle "from below-upwards". The derivation rules are applied differently than at derivation of sentences: right parts of rules are replaced with corresponding left parts. Process of analysis is realized as follows. On a first step all words of source chain are marked with strokes, i.e. replaced with corresponding chains with use of derivation rules of a kind $a' \to a$ (for instance, word *a* is replaced with chain *a'*). On the second step we look for subchains of a kind *a'b'* or *b'a'* in a chain, *ab* and *ba* – are marginal syntagms with a determined member *a*, and are replaced with chains of a kind *a'* with use of rules $a' \to a'b'$ (or $a' \to b'a'$). Then the second step repeats cyclically. Process of syntactical analysis is over when we receive start symbol *I* or chain that includes more than one symbol *I*. In the latter case the analyzed chain, by virtue of the statement 2, is not the sentence of language $L(F)$.

From the statement 2 and necessity of a lemma follows

The statement 3. *If* $\rho \in L(F)$ *– the any sentence, and ab* (*or ba*) *– its marginal syntagma*, *then chain* σ *recieved from* ρ *by removal of determining member b of syntagma ab* (*or ba*) *is the sentence of language L*(*F*).

The statement 3 provides receiving of the sentence of language *L*(*F*) after elimination of determining members of all unseparated marginal syntagms. According to this statement raising of syntagms by derivation rules of grammar *F* could be replaced by more effective cyclic process. On a first step of this process we look for unseparated marginal syntagms in the analyzed sentence. On the second step determining members are excluded from these syntagms. Then process repeats the same way till we get absolutely determined member as its single word in each sentence of the text.

The semantic analysis of text

Informativity of syntagmatic structures.

Informativity of syntagmatic structures could be evaluated with use of results of syntactical and statistical processing of text thematic corpuses Th_i and the full corpus of texts *Fu* [Липницкий, 2006].

Let us denote with *Sint* a set of all syntagmatic structures of the full corpus of texts *Tu*.

Let us examine the following population of events:

- S_{Th} – some syntagmatic structure α is taken randomly from the thematic corpus of texts *Th*;
- V_{Th} – syntagmatic structure α belongs to thematic corpus of texts *Th*;
- H_{Th} – occurrence of the thematic corpus of texts *Th*;
- S_{Fu} – syntagmatic structure α is taken from the corpus of texts *Fu*.

Let $P_\alpha(S_{Th} / S_{Fu})$ – is conditional chance that a syntagmatic structure α is taken from the thematic corpus of texts *Th* with the assumption that it is already taken from the full corpus of texts *Fu*. This conditional chance, as known, equals

$$P_\alpha(S_{Th} / S_{Fu}) = \frac{P(S_{Th} \cdot S_{Fu})}{P(S_{Fu})} = \frac{P(S_{Th}) \cdot P(S_{Fu} / S_{Th})}{P(S_{Fu})}.$$

Conditional chance $P_\alpha(S_{Th} / S_{Fu})$ we shall name *informativity* of the syntagmatic structure α in the thematic corpus of texts *Th*. If *Th* ($Th \subset Fu$) – is a text document, then we shall name this conditional chance informativity of the chain α in the text document *Th*.

We shall name syntagmatic structure α informative in the thematic corpus of texts (or in the text document) *Th* with *informativity level* p_0 if informativity of chain α is not less than p_0, i.e. $P_\alpha(S_{Th} / S_{Fu}) \geq p_0$.

Conditional chance $P_\alpha(S_{Fu} / S_{Th}) = 1$ as the event that syntagmatic structure α is taken from the full corpus *Fu* with the assumption that it is taken already from thematic corpus *Th* is authentic, because *Th* – is a subset of set *Fu*. Then we shall receive after simple transformations:

$$P_\alpha(S_{Th} / S_{Fu}) = \frac{P(V_{Th} / H_{Th})}{P(S_{Fu})} \cdot P(H_{Th}).$$

If we have a big enough full corpus of texts *Fu* and thematic corpus (or text document) *Th* then it is possible to consider

$$P(V_{Th} / H_{Th}) \approx \frac{n_{Th}}{N_{Th}},\ P(S_{Fu}) \approx \frac{n_{Tu}}{N_{Tu}},\ P(H_{Th}) \approx \frac{N_{Th}}{N_{Fu}},$$

where n_{Th}, n_{Fu} – are absolute occurrence frequencies of syntagmatic structure α in thematic and full corpuses of texts, and N_{Th}, N_{Fu} – quantity of all syntagmatic structures from set *Sint* in corpus *Th* and *Fu* accordingly. Then the formula for evaluation of informativity I_{Th}^{α} of syntagmatic structure α in the thematic corpus of texts (or in text document) *Th* will look like

$$I_{Th}^{\alpha} = \frac{n_{Th}}{n_{Fu}}.$$

Pragmatically full syntagmatic structures.

A pragmatically full syntagmatic structure (PF-structure) – it is a syntagmatic structure in a form of expression set that is informative in some thematic section of a subject domain (i.e. at least in one thematic corpus of texts).

Let us formalize concept of PF-structure. Let us consider some sentence $\pi = a_1a_2...a_{i-1}a_ia_{i+1}...a_n$ of source language $L(F)$, where $a_1, a_2,..., a_{i-1}, a_i, a_{i+1},..., a_n$ – are words of the sentence. Let a_i – is an informative word of the sentence. Consistently attaching to a word a_i at the left and on the right other words of the sentence π we shall form a set Ch_0 of all its 2-words, 3-words (and so on) subchains, wich syntactical graphs are oriented trees. Let us match each selected subchain α and probability $P(\alpha)$ of its occurrence in the full corpus of texts *Fu*. We shall choose limit value p_0 of this probability and we shall remove from set Ch_0 all chains with probability of occurrence in corpus *Fu* less than p_0. Let us denote with Ch_2 a set of all remaining 2-words chains in Ch_0, with Ch_3 – 3-words chains and so on. We shall denote with Ch_j ($j \geq 2$) nonempty set with maximal index and shall introduce next concept.

All subchains of chain π of set Ch_j we shall call *pragmatically full syntagmatic structures*.

Situational-syntagmatic network.

The semantic analysis of the text in abstracting system is realized by means of the subject domain knowledge base that is presented as situational-syntagmatic network i.e. the graph [Кравцов, 2006]. Nodes of the graph are syntagmatic structures, links are their situational bindings that are formalized as situational relation in a set of syntagmatic structures.

Let us denote with *Str* a set of all syntagmatic structures of the full corpus of texts *Fu*. Then tolerance relation Θ (reflexive and symmetric binary relation) on the set *Str* we shall name *situational relation* in the full corpus of texts *Fu* if any ordered couple of syntagmatic structures (μ, ν) of the set *Str* is an element of relation Θ if and only if probability of co-occurrence of structures μ and ν in the corpus *Fu* not less than a proper limit value (a *level* of situational binding). Saying co-occurrence of two syntagmatic structures we mean a presence of these structures in the same sentence of corpus *Fu*.

A graph of situational relation S_Θ we shall name a *situational-syntagmatic network*.

Route and graph of informativity of the text.

Let there is a text *T* (i.e. a tuple of sentences). We shall find out a subject of the text *T* and shall choose the corresponding thematic corpus of texts *Th*. (Depending on a task the subject of a text document could be found automatically by abstracting system or manually by user under the rubricator.) Let us calculate informativity of syntagmatic structures of all sentences of the text *T*. We shall use the full corpus of texts *Fu* and a thematic corpus *Th* for this purpose. We shall exclude from *T* all not informative sentences (i.e. sentences that have no informative structures). We shall get a tuple of sentences $T_{inf} = \langle \pi_1, \pi_2, \ldots, \pi_n \rangle$ in occurrence order in *T*. A tuple T_{inf} we shall name a *route of informativity* of the text *T*.

Let's build an oriented graph G_{inf}, assuming that all sentences of the route of informativity T_{inf} are nodes. Any pair of nodes π_i, π_j ($i < j$, $1 \leq i \leq n-1$, $2 \leq j \leq n$) – is a link (π_i, π_j) if and only if there is a pair of linked nodes (subchains of sentences π_i and π_j) in situational-syntagmatic network S_Θ. Link is showing that there is a situational binding between subchains.

An oriented graph G_{inf} with full oder on a set of nodes corresponding to sentences order in route of informativity T_{inf} we shall name a *graph of informativity* of the text *T*.

Semantic trace of the text.

A route of informativity T_{inf} is a basis for construction of the abstract of text T in a form of sequence of informative sentences. Let us build a semantic trace of the text to reduce quantity of sentences in the graph of informativity. Let us define a semantic trace as follows.

A *semantic trace Tr* of the text T is a subgraph of the graph of informativity G_{inf}, wich nodes are all nodes of the oriented graph G_{inf}, with quantity incidental links not less than n_0. Links of the oriented graph *Tr* are all links of the oriented graph G_{inf} that ties in *Tr* only adjacent nodes (figure 1).

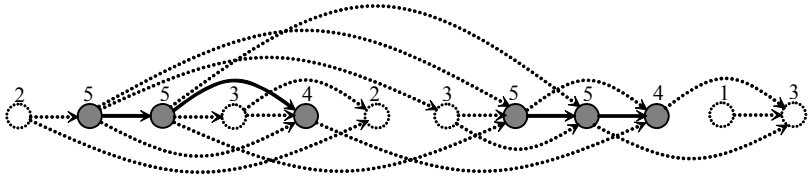


Fig. 1. An example of a semantic trace of the text in the graph of informativity

On figure 1 each node of the graph of informativity G_{inf} is marked with the number indicating quantity of incidental links. Nodes and links of the oriented graph G_{inf} that are not a part of the semantic trace *Tr* are shown as dashed lines.

A semantic trace of the text is a model of abstract that is constructed by abstract system.

Implementation in software

Visualization of informative sentences in the text.

On the basis of the offered model an experimental program of visualization of informative sentences in Russian text documents is developed [Насуро, 2006]. The program can process source documents in the following formats: html, txt, rtf, doc. These formats were chosen as they cover the majority of formats of available scientific-technical texts. The program can process pdf-documents by conversation to supported formats by third-party software such as paq pdf2txt [paqtol, 2006] or able2convert [able2convert, 2006] or others.

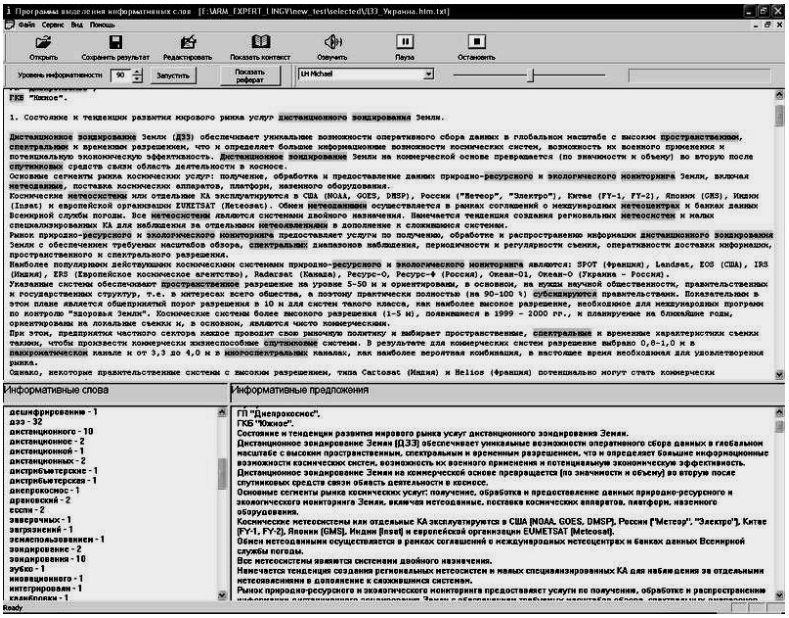


Fig. 2. The main window of the program of visualization of informative sentences

The experimental program is an executable application for Windows operating systems. The main window of the program is divided into three working areas. Top working area is for source file visualization. Left working area is for a list of found informative words. Working area at right side of the main window of the experimental program is for visualization of found set of informative sentences. Left and right working areas are empty by default and are filled with data when program processes a text document. The program has a number of toolbars. Service functions of the application are available from the program menu. These functions are as standard for windows applications then also specific for this program. They include file open, save and close operations, print results, help and settings of the program (figure 2).

The developed application has a toolbar “informativity level” that it used to set up a necessary level of informativity. A set of informative sentences will be formed from source sentences that include informative words with informativity equal or more than a defined level.

The program analyses quantity of words in the source document to chose an algorithm of search of informative words in big documents or in small texts. When program processes a big document using algorithm for small texts it selects the useful thematic information. The thematic corpus of texts is a kind of thematic filter in this case.

The program uses inflexion paradigm. It is very important especially for Russian that is an inflexional language.

Found informative words are highlighted. The program forms a file with selected informative sentences that could be edited and saved.

The program allows to see a context of the informative sentence that is highlighted in edit mode (figure 3).

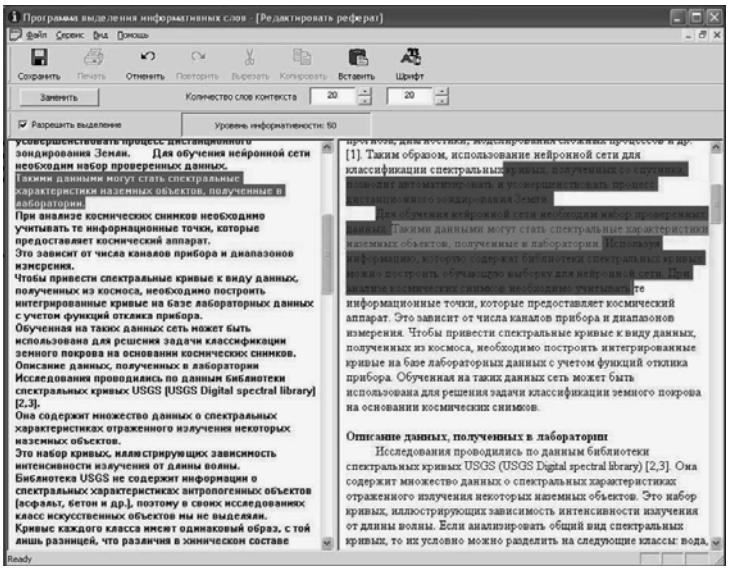


Fig. 3. The edit mode of the program

The user can adjust the length of context.

The program forms an information portrait of the document. The information portrait is a set of PF-structures most often used in the document and a set of informative words with occurrence frequencies. The program allows to adjust quantity of displayed PF-structures and informative words.

When process a document the program uses statistical data from an inflexion paradigm dictionary. This dictionary is formed from thematic corpuses of texts. Thematic corpuses of texts are formed with special software that is developed in the United Institute of Informatics Problems of the National Academy of Sciences of Belarus. Data obtained as a result of the processing is a knowledgebase of the experimental program of informative words visualization.

The program is developed on C++ programming language.

Conclusion

The offered mathematical model of text documents analysis allows to realize detection of informative sentences in big texts.

Monothematic fragments of the text document could be detected in a set of informative sentences with use of situational-syntagmatic network.

A complex of special software is developed on the base of results of mathematical modeling. The software is for automatic abstracting of Russian texts. The size of the abstract can be adjusted by setting up a syntagma informativity level and text semantic trace characteristics.

Bibliography

[Кравцов, 2005] Кравцов А.А., Липницкий С.Ф., Насуро Д.Р., Прадун Д.В. Интеллектуализация процессов обработки текстовой информации // Информатика. – 2005. – № 1. – С. 41–51.

[Липницкий, 2005] Липницкий С.Ф. Семантический анализ текста на основе ситуативно-синтагматической сети // Информатика. – 2005. – № 2. – С. 102–110.

[Липницкий, 2006] Липницкий С.Ф. Математическая модель и алгоритмы формирования схемы грамматики, порождающей проективные предложения // Весці НАН Беларусі. Сер. фіз.-тэхн. навук. – 2006. – № 3. – С. 71–75.

[Кравцов, 2006] Кравцов А.А., Липницкий С.Ф., Насуро Д.Р. Синтез рефератов текстовых документов на основе ситуативно-синтагматической сети // Искусственный интеллект. – 2006. – № 2. – С. 172–175.

[Насуро, 2006] Насуро Д.Р. Алгоритмы и программы визуализации информативных предложений в системе автоматического реферирования текстовых документов // Искусственный интеллект. – 2006. – № 2. – С. 416–419.

[paqtol, 2006] A web site of pdf-conversion tool. – http://www.paqtool.com

[able2convert, 2006] A web site of pdf-conversion tool. – http:// www.investintech.com/prod_a2e_pro.htm

Authors' Information

Stanislav F. Lipnitsky – *Doctor of Sciences, United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova str. 6, Minsk, 220012, Belarus; e-mail: lipn@newman.bas-net.by*

Denis R. Nasuro – *Research fellow, United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova str. 6, Minsk, 220012, Belarus; e-mail: nasuradr@newman.bas-net.by*

LEXICON OF COMMON SCIENTIFIC WORDS AND EXPRESSIONS FOR AUTOMATIC DISCOURSE ANALYSIS OF SCIENTIFIC AND TECHNICAL TEXTS *

Elena Bolshakova

Abstract*: Various NLP applications require automatic discourse analysis of texts. For analysis of scientific and technical texts, we propose to use all typical lexical units organizing scientific discourse; we call them common scientific words and expressions, most of them are known as discourse markers. The paper discusses features of scientific discourse, as well as the variety of discourse markers specific for scientific and technical texts. Main organizing principles of a computer dictionary comprising common scientific words and expressions are described. Key ideas of a discourse recognition procedure based on the dictionary and surface syntactical analysis are pointed out.*

* Work is supported by the grant 06-01-00571-a of Russian Fond of Fundamental Researches.

***Keywords**: scientific discourse, discourse markers, common scientific words and expressions, scientific discourse operations, discourse-compositional analysis of scientific and technical texts.*

***ACM Classification Keywords**: I.2.7 [Artificial Intelligence]: Natural language processing – Text analysis*

Introduction

Functional style of scientific and technical prose is admittedly the most distinctive one, primarily due to intensive use of scientific phraseology and structuring. The phraseology includes, besides scientific and technical terms of a specific terminology, various expressions of common nature, such as English expressions *exploratory study, mentioned above, for this reason, in addition, therefore,* etc. and *Russian: вышеупомянутый, по этой причине, в дополнение к, далее мы опишем* and so on. We call such lexical items ***common scientific words and expressions***.

Within scientific discourse (scientific speech), terms and common scientific expressions differ in their functions. Specific terms denote concepts, objects, and processes of the particular scientific domain, whereas common scientific expressions are domain independent: they are used to design and organize scientific text narrative by expressing the logic of reasoning, by connecting text fragments devoted to different topics and subtopics, and by structuring the text under development.

Similar to terms, common scientific expressions present a syntactically quite heterogeneous set, and to an even greater degree than terms. The set comprises, besides content (autosemantic) words, functional (auxiliary) words. Noun and verb-noun combinations, adverb and participle expressions, compound prepositions and conjunctions are included as well. Among the word combinations, one can notice stable expressions exploited as ready-for-use colloquial formulas (clichés) [1], such as *Eng. as it was stated above*, *to outline directions of further research; Rus. из вышесказанного следует, как показало проведенное исследование*. Some clichés are common for scientific and technical prose, the others are specific for particular genres. Certain common scientific words and expressions are known as discourse markers [11, 12].

The paper reports on preliminary results of an ongoing research aiming at elaboration of a procedure for discourse analysis of scientific texts, as well as development of an adequate computer dictionary of common scientific words and expressions. This is done within overall framework of creating computational models of scientific and technical prose. Our basic claim is that in order to attain a really effective automatic processing of scientific and technical texts (which is needed for many applications, in particular, for text summarizing [5]), we should take into account functional peculiarities of scientific prose on various levels, in the first place, levels of phraseology and discourse.

The research involved an empirical study of scientific texts in several fields of exact and natural sciences, so that scientific papers as the core of the functional style were analyzed. The study was initially performed for Russian texts, and then expanded to English. As the work progressed, the importance of the common scientific lexicon became increasingly obvious, despite of its relatively small size. In both languages the principal features of scientific lexicon and discourse proved to be the same, which emphasizes the international character of scientific and technical prose.

Advancing towards an appropriate procedure of discourse analysis, we create at the first step a computer dictionary that comprises a wide range of common scientific words and expressions and provides a classification of their syntactic and semantic features. For Russian, the dictionary is now partially implemented; for English, only the classification work was done so far.

In comparison with DiMLex lexicon of discourse markers [11], which was developed for German and English and mainly consists of conjunctions and conjunctive adverbs, our dictionary covers a wider set of lexical units, because we consider any lexical device signaling scientific discourse as its marker (e.g., English expression *by definition* or Russian *по определению*).

A discourse-analyzing procedure is also under development now. Since units of common scientific lexicon may be served as surface cues, we assume the hypothesis that shallow text analysis based on the lexicon is adequate for detecting discourse structure of scientific text, without a deeper syntactical-semantic analysis of all its sentences. Instead of concept of discourse relationship, which is proposed in well-known discourse theory RST [7] for explaining relations between adjacent phrases in text, we rely on the concept of *scientific discourse operation* for recognition of discourse-compositional text structure specific for scientific texts. Our discourse recognition procedure also differs from the procedures that were developed for Japanese texts [6, 9] and based on deep syntactical analysis of sentences, with consideration of style-independent discourse markers.

The objectives of this paper are:

- To determine the set of common scientific words and expressions we regard as discourse markers;
- To describe designing principles for the corresponding computer dictionary;
- To sketch the procedure for recognition of discourse-compositional structure of scientific texts;
- To point out potential application of the dictionary and the procedure being design.

To clarify our ideas we begin with an overview of specific features of scientific discourse, which are derived from our empirical study. The features proved to be language-independent, and we give in the paper illustrative examples from both languages: English and Russian.

Scientific Discourse and Its Devices

Discourse consists of interrelated speech acts determined by communicative goals. The global purpose of scientific communication is to convey new ideas and results of scientific research, as well as to explain and rationalize them. Therefore, scientific discourse involves reasoning that is organized as a sequence of mental operations of informing and arguing. Among typical operations we should point out assuming hypotheses, defining new terms, determining causal relations, exemplification, resuming and so on. We will call such intellectual operations ***scientific discourse operations***.

As a tendency of scientific and technical prose to be strict and plain, these discourse operations are usually introduced into texts and more or less explicitly marked by authors of texts with the aid of lexical devices – common scientific words and expressions. With this function, the words and expressions pertain to metatext component of discourse [12] and are called ***discourse markers***.

We consider scientific text as composed of ***discourse segments***, each segment including several adjacent sentences and corresponding to the applied discourse operation. Some sentences include discourse markers.

The most evident markers of scientific discourse are ***mental performative expressions***, or ***performative formulas*** like *Eng. we conclude, we would assume* or *Rus. мы докажем, мы предположим*. For Russian, they are described in detail in [10]. Performative formulas are based on "mental" verbs, e.g., *Eng. to conclude, to consider, to admit, to propose; Rus. заметим, рассмотрим, выразим* and so on. As a rule, these verbs explicate particular steps of scientific reasoning and have valences (complementing arguments): *Eng. we consider N*, *we conclude that S* ; *Rus. рассмотрим N*, *подчеркивается, что S*. Besides pure "mental" verbs (*to conclude, to assume,* etc.), verbs of physical action (*to see, to show,* etc.) are used as mental. The class of such verbs is open, since various verbs may be potentially used as mental in the context of scientific discourse.

There are various forms of mental performative expressions in scientific texts:

- Canonical forms, with mental verb in the second person plural, often with the corresponding pronoun (e.g., *Eng. we resume, let us proceed to, we will proceed; Rus. мы покажем, мы рассмотрим*);
- Verbal variants (*Eng. summing up, strictly speaking; Rus. подводя итоги, строго говоря*), which are often used together will canonical forms (*Eng. refining the definition, wee see that…; Rus. суммируя вышесказанное, укажем…*);

- Impersonal forms (*Eng. it should be added, it was found, it is reasonable to assume; Rus. необходимо/нетрудно заметить, представляется, что..*), which often include words of author's estimation (*should, reasonable, necessary*);

- Descriptive variants (*Eng. N is briefly described, N are given in; Rus. N кратко описано*)

Verbal and impersonal forms are used in texts to paraphrase canonical forms (e.g., *it was found* instead of *we found*) or to give some cross-text references (e.g., *as it was stated above*). Though they are less explicit forms than canonical, they are functionally equivalent. One can also find 'hidden' performatives in scientific and technical texts, which we call descriptive variants: e.g., *These data are given in Table 3* stands for *We gave these data in Table 3*.

Mental discourse operations might be expressed by various parenthetic words and expressions: indicators of order (e.g., *Eng. first* or *lastly; Rus. во-первых, наконец*), markers of equivalency (e.g., *Eng. in other words; Rus. иными словами*), various connectives between textual parts (e.g., *Eng. nevertheless* or *so far; Rus. тем не менее, благодаря тому, что*) and so on. The metatext nature of these discourse markers is more obvious, they are typical not only for scientific and technical texts.

Among words typical for scientific discourse we should mention abstract nouns, such as *problem*, *analysis*, *model, concept, conclusion* and so on. They are aimed to name mental constructs by which scientific information is semantically structured. We call such nouns ***common scientific variables***, since they have the obligatory semantic valence (*problem of N, model of N*). Common scientific variables are mainly used with mental performative verbs, thereby forming stable noun-verb combinations, such as *Eng. to test hypothesis* or *to draw conclusions; Rus. подвергнуть анализу, проводить аналогию, опровергнуть гипотезу* [4]. Meanings of such verbs are close to Mel'čuk's Lexical Functions [8] with corresponding nouns as arguments.

Below we present several English text fragments from the book on artificial intelligence and from the papers [5,11], which illustrate the usage of common scientific words and expressions (they are underlined):

> *In fact, notice that the value of the slot Players is a set. Suppose, instead, that we want to represent the Dodgers as a class instead of an instance. … For example, we could make it a subclass of major league baseball players.* ***(1)***
>
> *According to our corpus study, we have identified three basic rhetorical configurations for summaries, that we call Meta-Schemas.* ***(2)***
>
> *For dealing with discourse markers, we do not regard this distinction as particular helpful, though. As we have illustrated above and will elaborate below, these words can carry a wide variety of semantic and pragmatic overtones, which render the choice of a marker meaning-driven, as opposite to a mere consequence of structural decisions.* ***(3)***

Besides common scientific lexicon, non-lexical devices are used to organize scientific discourse. In particular, such devices as sections, paragraphs, items, rubrics, and numeration are intended to structure scientific texts and to form their composition. All structuring and discourse-organizing devices present an interconnected system: devices can complement or substitute one another. For example, section headings are really substitutes for performative expression *we proceed to*, whereas numeration often complements performative formulas: e.g. *Let us enumerate main statements: 1)…2)…* . This interconnected system is rather excessive, since for most discourse operations there exist collections of similar lexical markers, but at the same time it allows for flexible paraphrasing.

In general, some discourse operation with its lexical and non-lexical devices can be used to implement another operation. For example, for categorization, a definition of new term is often required. Therefore certain discourse segments are embedded into some others, and in this way hierarchical structure of scientific text is formed.

Computer Dictionary of Common Scientific Lexicon

To develop a computer dictionary, collections of Russian and English common scientific words and word combinations were gathered from few available text dictionaries of scientific phraseology [3, 4] and from scientific texts in several fields of science (mainly in computer science and artificial intelligence), through their manual scanning. While selecting a word or expression for our collection, we used the following non-formal criteria. First, discourse-organizing function of the word or expression should be evident, second, it should be rather frequently used in texts in several fields. Inter-language correspondences were used: for Russian expressions English equivalents were looked for, and vise versa.

Heterogeneous collections of words and word combinations were classified according to their discourse-organizing functions in scientific texts, irrespectively of their grammatical form and syntactic features. Based on our study, we propose for classification the list of scientific discourse operations, the most significant are given in Table 1:

Table 1. Scientific discourse operations

Operation	Russian Examples	English Examples
Description or statement	*укажем, что; характеризуя*	*let us to describe; we point out that*
Elaboration or adding information	*в частности; в дополнение к*	*to be more precise; in addition*
Expressing relations of causal, conditional, and concession type	*по этой причине; следовательно*	*hence; provided that; however*
Actualization of the topic	*перейдем к; рассмотрим*	*as for; let us consider; regarding*
Emphasizing	*особо подчеркнем;* *необходимо отметить*	*first of all;* *it is necessary to emphasize*
Presupposition	*предположим/допустим, что*	*we would assume; it may be admitted*
Definition	*будем называть; по определению*	*by definition; we call it/them,*
Comparison	*по сравнению с*	*as compared with*
Contraposition	*с одной стороны*	*on the one hand; as opposite to*
Illustration or exemplification	*к примеру; например*	*as illustrated below; for example*
Generation or resuming	*суммируя вышесказанное; в общем*	*in general; summing up*
Enumeration or ordering	*во-первых; наконец*	*next; finally*
Labeling with a scientific variable	*идея; модель; результат*	*result; idea; model*
Expressing of author's attitude	*целесообразно считать;* *по всей видимости*	*in our opinion*; *it seems reasonable*

Our collection of common scientific words and expressions was divided into functional classes in accordance with the proposed list of discourse operations. Within each class, all words and word combinations that are semantically close and interchangeable in the texts as discourse markers were gathered into a group, thereby giving a subclass of functionally equivalent markers. Each group of functional equivalence often includes words of different parts of speech and contains from 2 to 9 units, the number depending on the language. For example, the resulted group of the consequence relationship includes for English: *hence, therefore, as a result, consequently, it follows that, we conclude that* etc., and for Russian: *значит*, *итак*, *таким образом*, *тем самым, как видим* etc. For both English and Russian, we obtain 53 groups corresponding to particular discourse organizing functions.

To determine lexical entries of our computer dictionary, we considered requirements for its use by automatic text processing system, first of all, by discourse-analyzing procedure. The dictionary contains:

- Units corresponding to words of common scientific lexicon. They comprise both functional and content words, including those encountered only within scientific expressions.

- Units corresponding to common scientific word combinations.

For a particular word, each unit stores adequate morphosyntactic information, including the part of speech and the flexional class (if any), as well as pointers to dictionary units describing available combinations with this word.

In turn, each unit for a particular word combination accumulates necessary syntactical properties of the combination: stable *vs*. free, continuous *vs*. discontinuous. Since most word combinations have syntactic valences, we propose to represent information about valences with the aid of special ***lexicosyntactic patterns***.

Each lexicosyntactic pattern fixes lexemes (constituent words of the particular combination) and their grammatical forms, as well as specifies syntactic conditions necessary for filing its empty slots (valences of the fixed lexemes). An example of such a pattern is *"let us consider" NP* with *NP* denoting a noun phrase. Another example is *NP "we will call" T*, where *T* denotes an author's term and *NP* is a noun phrase explaining its meaning; it describes the typical English expression for definition of new terms.

A formal language for specifying lexicosyntactic patterns was elaborated, as well as a methodology for acquiring new patterns for the particular discourse operation from scientific and technical texts. Lexicosyntactic patterns proved to be a convenient device for describing stable colloquial expressions comprising both phrasal formulas (like *the paper describes main features of*, *argument can be made against*) and predicative constructs (such as *to take as starting point for*). Based on the acquiring methodology, a collection of patterns was created, which describes typical Russian single-sentence definitions of new terms. For example, one of lexicosyntactic patterns for discourse operation of defining a new term is

«под» NP1 <case=ins> V<пониматься; tense=pres, person=3> NP2 <case=nom> < NP1.numb=V.numb>

Particular lexemes of the pattern are quoted, letter *V* denotes the verb, NP1 and NP2 denote noun phrases, and grammatical conditions are written within angle brackets – they specify values of grammatical parameters (tense, person, case, number) or establish their equality. The pattern describes both Russian sentences *«Под графемной конструкцией понимается графическая форма, построенная из базисных, проблемно-ориентированных и/или графических конструкций»* and *«Под данными при такой формализации понимаются последовательности символов в некоторых алфавитах»* (in these sentences fixed lexemes of the pattern are underlined) .

In addition, for each dictionary unit considered as discourse marker, the developped computer dictionary provides semantic information that facilitates recognition of underlying discourse operation, namely:

- Functional class and group of the unit within the proposed semantic classification;
- Contextual conditions necessary for being discourse marker within texts;
- Information about size and boundaries of implied discourse segment (the segment consists of one sentence or of several sentences; the dictionary unit marks the beginning or the end of discourse segment).

Discourse-Analyzing Procedure

Our study of scientific discourse showed that common scientific lexicon has its own functional semantics, which makes it possible to superficially read scientific texts, i.e. to derive underlying discourse operations and to comprehend logic of scientific reasoning, without deep understanding of these texts. So we are developing our procedure for recognition of discourse-compositional structure of scientific texts on the basis of shallow text analysis and the described computer dictionary.

We consider discourse-compositional structure of scientific text as hierarchical structure of sequenced and embedded discourse segments, which corresponds to applied discourse operations and applied structuring devices. The structure may be represented as a tree, with tree nodes corresponding to discourse segments, and tree links fixing semantic (in particular, causal) and structural (in particular, embedding) relations between segments. In order to construct such a tree for a given text, the proposed recognition procedure takes the following steps:

1. Grapheme analysis of words, delimiting of sentences, and detecting of text composition elements: section headings, paragraphs, items, rubrics, and numeration.
2. Morphologic analysis of words and identification of occurrences of common scientific words and word combinations.
3. Recognition of dictionary discourse markers in the given text through matching text fragments with dictionary lexicosyntactic patterns that contains identified common scientific words.
4. Delimiting of discourse segments, based on recognized discourse markers and semantic information presented in the dictionary for functional groups and classes. In general case, the result of the segmentation is ambiguous, since several plausible discourse trees fit the sequence of identified markers.
5. Selection of the most plausible discourse-compositional tree within the set resulted at the previous step. A number of heuristic rules are used for this purpose, for example, an exemplifying segment is rather embedded into another segment than embeds it.

To implement steps 3 and 4 surface syntactical analysis of sentences is needed, which takes into account:

1. agreement and coordination of words;
2. overall grammatical structure of sentences.

It should be noted that reliability of discourse recognition depends on several factors, among them are the number and types of discourse markers encountered in the text. In order to increase the reliability, the other linguistic devices, in particular, anaphoric links and repetitions of lexical units in adjacent sentences are to be considered.

Conclusion

We have overviewed the features of scientific discourse and the spectrum of common scientific words and expressions, with their role in scientific discourse. We described main organizing principles of the computer dictionary of common scientific lexicon, which provides various information valuable for automatic analysis of scientific and technical texts. We have also outlined heuristic multi-step procedure for recognition of scientific discourse-compositional structure, with the aid of the dictionary and surface syntactical analysis of sentences.

The recognized discourse markers and discourse-compositional structure is apparently useful in computer systems intended for

- Text abstracting, which may be based on processing of detected markers, e.g. *we illustrate our approach with N* transforms into *the approach is illustrated with N*;
- Document browsing and intra-document information retrieval, which are especially topical for large-size technical documents;
- Computer-aided writing and editing of scientific and technical texts;
- Eliciting of knowledge represented in scientific and technical texts, in particular, extraction of new terms and their definitions introduced into a text by authors.

These applications will supposedly be investigated after implementation, testing, and refinement of the dictionary and the recognition procedure. But more actual task now is creating of computer-aided procedures for enlarging the dictionary, in order to accumulate a comprehensive set of common scientific words and expressions.

Bibliography

1. Bolshakova, E.I. Phraseological Database Extended by Educational Material for Learning Scientific Style. In: ACH/ALLC 2001: The 2001 Joint Int. Conference. Conf. Abstracts, Posters and Demonstrations, New York, 2001, p. 147-149.
2. Bolshakova, E.I., Vasilieva N.E., Morozov S.S. Lexicosyntactic Patterns for Automatic Processing of Scientific and Technical Texts. In: Proc. of 10th National Conference on Artificial Intelligence with International Participation 2006. Moscow, Fizmatlit, Vol 2, 2006, p. 506-514 (in Russian).

3. Dictionary of Word Combinations Frequently Used in English Scientific Literature. Nauka Publ., Moscow, 1968.
4. Dictionary of Verb-Noun Combinations of the Common Scientific Speech. Nauka Publ., Moscow, 1973 (in Russian).
5. Ellouze, M., Hamadou A.B. Relevant Information Extraction Driven with Rhetorical Schemas to Summarize Scientific Papers. In: Advances in Natural Language Processing. Third International Conference, PorTAL 2002. E. Ranchhod and N.J. Mamede (Eds.). Lecture Notes in Computer Science, N 2389, Springer-Verlag, 2002, p. 111-114.
6. Kurohashi, S., Nagao M. Automatic Detection of Discourse Structure by Checking Surface Information in Sentences. In: COLING 94 Proceedings of the 15th Int. Conf. On Computational Linguistics. Vol. II, Kyoto, Japan, 1994, p. 1123-1127.
7. Mann, W.C., Thompson S.A. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. Text, 8 (3), 1988, p. 243-281.
8. Mel'čuk, I. Dependency Syntax: Theory and Practice. SONY Press, NY, 1988.
9. Ono, K., Sumita K, Miike S. Abstract Generation Based on Rhetorical Structure Extraction. In: COLING 94 Proceedings of the 15th Int. Conf. On Computational Linguistics. Vol. II, Kyoto, Japan, 1994, p. 344-348.
10. Ryabtseva, N.K. Mental Performatives in Scientific Discourse. Voprosy yazykoznaniya, V 4, 1992, p. 12-28 (in Russian).
11. Stede, M., Umbach C. DiMLex: A Lexicon of Discourse Markers for Text Generation and Understanding. Proceedings of Int. Conf. On Computational Linguistics COLING-ACM'98, Vol. 2, 1998, p. 1238-1242.
12. Wierzbicka A. Metatext w tekscie. In: O spojnosci tekstu. Wroclaw-Warszawa-Krakow-Gdansk, 1971, p.105-121.

Authors' Information

Elena I. Bolshakova *– Moscow State Lomonossov University, Faculty of Computational Mathematics and Cybernetic, Algorithmic Language Department; Leninskie Gory, Moscow State University, VMK, Moscow 119899, Russia; e-mail: bolsh@cs.msu.su*

TECHNOLOGY OF STORAGE AND PROCESSING OF ELECTRONIC DOCUMENTS WITH INTELLECTUAL SEARCH PROPERTIES

Yuri Kalafati, Konstantin Moiseyev, Sergey Starkov, Svetlana Shushkova

Abstract*: The technology of record, storage and processing of the texts, based on creation of integer index cycles is discussed. Algorithms of exact-match search and search similar on the basis of inquiry in a natural language are considered. The software realizing offered approaches is described, and examples of the electronic archives possessing properties of intellectual search are resulted.*

Keywords*: dynamic systems, associative search, integer index cycles, text indexation, archiving and retrieval.*

Introduction

The use of dynamic systems for record and data processing was first offered in [Dmitriev etc., 1991] . The basic idea consists in the fact that a correspondence is put between a given set of information blocks and a set of limiting cycles of a discrete nonlinear dynamic system: one-dimensional [Dmitriev, 1991], [Andreyev etc., 1992], [Andreyev etc., 1997] or multidimensional [Andreyev etc., 1994] maps. The final sequence of symbols of a certain alphabet is meant by the information block and each symbol is put in conformity with the dynamic system variable. This approach has been realized to record the text and graphic information by means of piecewise-linear maps and the scope for associative searching of information by its fragments is shown. The research has been developed further [Andreyev etc., 1999] and to optimize the search an integer index was suggested to be used instead of dynamic variables. In spite of the fact that the principles of information representation with the integer index cycles are well-known and successfully applied for data storage in various DB, the creation of electronic

archives possessing the opportunities of intellectual search, including associative search, exact-match search, search similar to inquiries in the natural language is of special interest and is to be developed. Consider some key moments of a technology being developed.

Indexation algorithm

Assume, the first page presented by the next text line is transmitted for indexation:

a b c a c a d a b a c a b c (1)

As a first step, the symbol – an attribute of the beginning of a page is added to the initial alphabet. This symbol will have the number 256. Transform the initial text to a cyclic sequence of symbols and add a symbol 256 to the end of a sequence of symbols. To avoid excessive information, the concept of an expanded alphabet is entered in the initial sequence. Each element of an expanded alphabet is based on two already existing symbols. For example, if it is necessary to escape the repeated sequence of symbols a b c, a new symbol 257 = ‘ a ’ + ‘ b ’ and 258 = 257 + ‘ c ’ is entered and so the sequence a b c in the initial text is replaced with a symbol 258. The text (1) is indexed using the above concept. Let us search the repeated pairs of symbols in the initial text. Take the pair of symbols from a given example– the pair a b. As a given pair of symbols is found more than once in the text, we add this pair to the expanded alphabet:

Symbol 257 = ‘ a ’ + ‘ b ’

Transform the initial sequence of symbols, having replaced the pair symbols a b by a new symbol with the index 257. We have:

257 c a c a d 257 a c 257 c 256 (2)

Then find the recurrence of other pairs and create new symbols. Based on this sequence, create a page description array, each element of which consists of three symbols: a_n, a_{n+1}, a_{n+2} , and order the array obtained in a pair (a_n, a_{n+1}). A necessary condition for system operation is the uniqueness of a pair (a_n, a_{n+1}) in the array formed. The following actions are to be done in adding the next page:

The text transmitted for indexation is processed using the expanded alphabet formed earlier. First there is a search in the added text of a pair of symbols corresponding to the element 257 in the expanded alphabet. If such pairs are found, they are replaced by the symbol 257. Then there is a search of a pair of symbols corresponding to the element 258 and so on.

The repeated pairs of symbols are searched in the sequence of symbols obtained. If these are found, a new element of the expanded alphabet is created and the corresponding pairs are replaced with a new element.

Then each pair of symbols from the sequence created is being searched in the page description array available. If such a pair is found, a new symbol of the expanded alphabet is created and the corresponding pair of symbols in the added text is replaced by a new symbol. In the program realization available there also proceed the change in already existing array and the replacement of a pair of symbols with the newly created symbol. The description of this algorithm updating is beyond the scope of this paper.

Transform the obtained cyclic sequence to a set of elements for a page description array. Add these elements to an array and reorder it.

The expanded alphabet is presented as a two-dimensional integer array. An array index is the number of a symbol in the expanded alphabet. The first element of an array has the index 257. Each couple of integers stored in the array element are components of the corresponding alphabet symbol. The ordered page description array is the array consisting of elements (a_n, a_{n+1}, a_{n+2}). This array is ordered in a pair (a_n, a_{n+1}) and this pair is unique within the whole indexed text *sets.* Input points *on indexed pages.* Any pair of symbols of each page representation is stored as a two-dimensional array of integers. Array index is the number of the page indexed, and the array element content is the information necessary and sufficient to begin the procedure of extracting corresponding text page.

Text extraction from the page

To extract the text page from the constructed index it is sufficient to have information about any pair of symbols in the expanded alphabet (a_{n0}, a_{n0+1}) contained in the page. It should be remembered that the page description array is ordered in the first pair (a_n, a_{n+1}) and the condition required. i.e. the uniqueness of this pair in the whole file index , is fulfilled. Using the procedure of binary search we are searching the pair (a_{n0}, a_{n0+1}) in a page description array. From the element found in a file we take an element a_{n0+2}. Then in the page description array we search the pair (a_{n0+1}, a_{n0+2}). From the element found in a file we take the element a_{n0+3} and so on. We repeat this operation until after search of the pair (a_i, a_{i+1}) and extraction of the element a_{i+2} the equality: $a_{i+1} = a_{n0}$ and $a_{i+2} = a_{n0+1}$ is true.

Thus, we have obtained the cyclic sequence describing the page text in symbols of the expanded alphabet. Then based on the procedure of decoding the symbol of then expanded alphabet (see below) we transform the cyclic sequence presented in symbols of the expanded alphabet in a sequence of ASCII symbols.

Description of base algorithm search

To demonstrate the principal opportunity of information search with a given way of information indexation the following mechanism is suggested. A sentence or even a text paragraph is taken as an input for searching. The search inquiry is coded by means of the expanded alphabet available. Assume that the search inquiry was transformed to the sequence a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 of elements of the expanded alphabet. Take the first pair of the search inquiry (a_1, a_2) and search for it in the ordered page description array. If such a pair of symbols is not found, pass to the pair of symbols (a_2, a_3) and so on. In case of a successful search, for example, of the pair (a_2, a_3), we have (a_2, a_3, a_x) an element of the page description array. Compare the element a_4 and the element a_x. If these elements coincide, find (a_3, a_4, a_y) the element of a page description array. If elements a_5 and a_y coincide, the search is considered to be successful.

Two versions of search are realized based on the suggested indexation algorithm of text information:

Exact-match search

A given version of search is intended for solving a problem of search by a word or by a short search inquiry. It allows to find all inputs of search inquiry in the indexed text array. According to the offered algorithm of indexation there are some possible variants for line representation corresponding to the search inquiry inside the indexed file. First, the whole line of a search inquiry can be entered inside a symbol of the expanded alphabet. All such symbols as well as symbols created on their basis are to be searched in a page description array. Each found element of a page description array which is the input point to one of the pages of the file indexed is considered to be the result of search.

Second, the line, corresponding to the search inquiry in the indexed page, can be divided into two parts, one of which is the end and the second is the beginning of one of symbols of the expanded alphabet. To find all such results of search we should construct two files of symbols. The first one is an array of all symbols of the expanded alphabet terminating with a given line of ASCII symbols. The second one is an array of all symbols of the expanded alphabet beginning with a given line of ASCII symbols. The corresponding pair of symbols should be searched for each element of the first array and each element of the second array in the page description array. In successful search the found element of a page description array is a result of search. Then the required line can consist of three parts. The first part is the end and the third one is the beginning of one of the symbols of the expanded alphabet. The second part is one of the elements of the expanded alphabet. As in the previous case, compile two files of symbols for the first and third parts of the search inquiry. For each symbol of the first part we create a pair of symbols formed by an element of this array and a symbol of the expanded alphabet corresponding to the second part of the search inquiry and search this pair in a page description array.

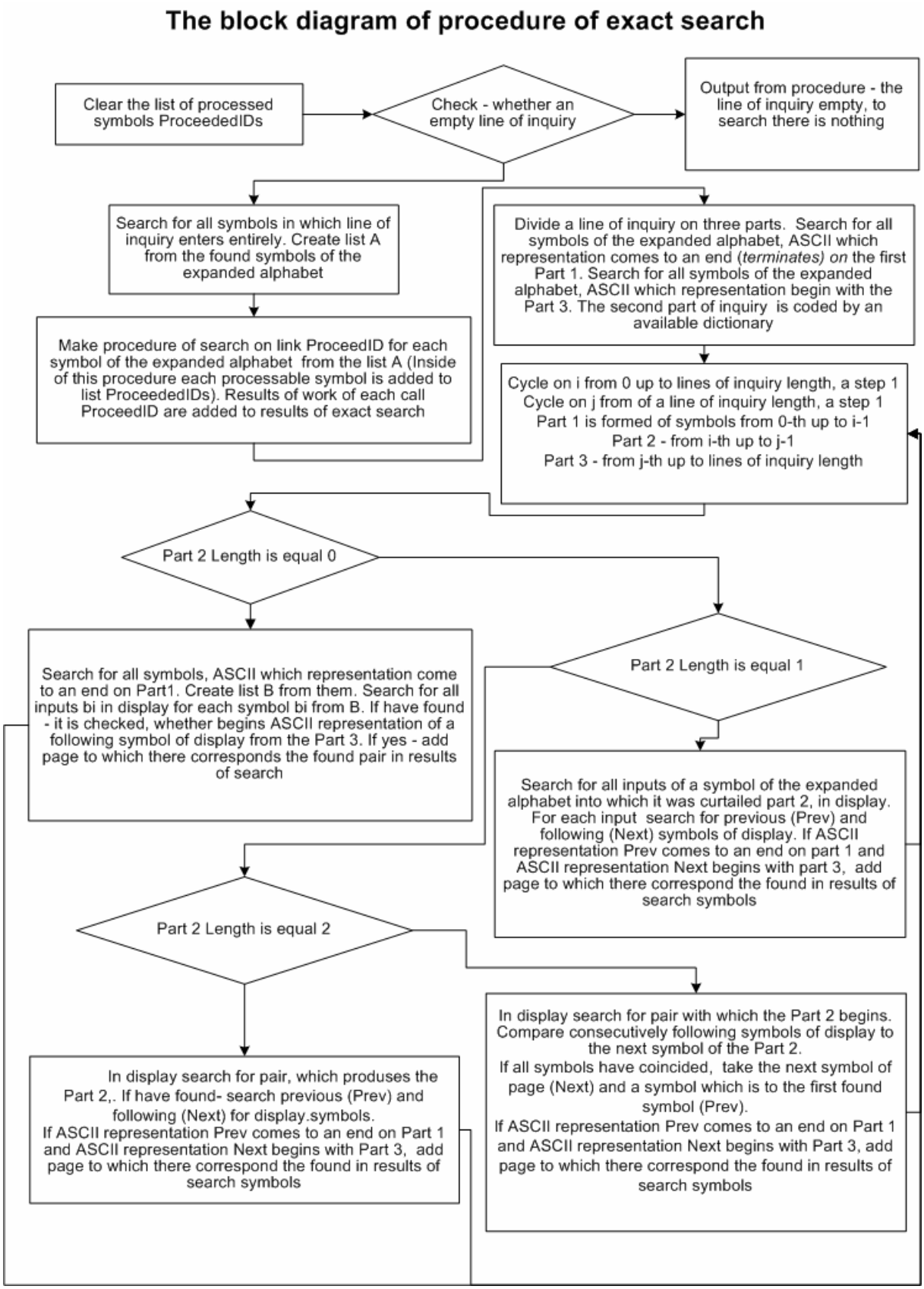


Fig. 1. Procedure of exact-match search.

If search is successful, search the third symbol of a file element in the file of symbols constructed for the third part of search inquiry. In case of success the found element of a page description array is considered to be the result.

Finally, if the number of parts by which a desired line in the indexed page is presented exceeds three, we can search the symbols displaced in the centre of the line. Assume that the line is presented by five parts. The first and fifth parts are, accordingly, the end and the beginning of symbols of the expanded alphabet. Each of the remaining parts is presented by one of the symbols of the expanded alphabet. First, we search for the pair of the symbols formed by the second and third parts. If this part is found in a page description array, the third symbol of this array element can be compared with the fourth part. If they coincide, a pair of symbols formed by the third and the forth parts is being searched in a page description array. Take the third element from the found element of a file and we search it in the fifth part, i.e. in a file of symbols beginning from with the end of search inquiry. If search is successful, it is necessary to compare only a part of the found page which is before the second part of a line with the first part. In case of success we have the result of search. Fig.1 presents a block-diagram of the exact search procedure.

Search of similarity

This variant of search is meant for finding information closest to the search inquiry. Offer in the natural language, a paragraph or even the whole page of the text can be transmitted as the search inquiry. The search inquiry transferred to the input of search of similar is coded by means of the expanded alphabet available.

On the basis of a list of symbols for each indexed page the following sum is calculated:

$$P_i = \sum_{k=1}^{N} (length_k)^{\alpha} * (count_k)^{\beta} ,$$

where $length_k$ is the length in ASCII symbols of the k-th element in a list of symbols, $count_k$ is the quantity of the k-th element in a list in page i, and are the external parameters. Then the obtained Pi values are ordered and pages with the highest values are given to the user as results of search.

Conclusion

Now a described algorithm of text processing and algorithms of text-through search are realized and used in CCT Archive and CCT Publisher Companies Controlling Chaos Technologies software products. Software products are intended for the creation of electronic archives of not structured documents with an opportunity of text-through information search, and for creation and preparation for CD and DVD electronic books, encyclopedias, archives of magazines. Examples of successful application of software products are the electronic archives of magazines « Chemistry and the Life », "Quantum", "Knowledge-force".

Fig. 2. gives results of search system operation with electronic archive of magazine " Quantum " as an example. At the upper left is inquiry in the natural language on which the search was carried out, below is the ranged list of the documents found. To the left is the document page with the allocated inputs.

Below are the basic time characteristics managed to be reached with the present program realization of the algorithms described. All values are obtained using an ordinary personal computer, by the text size we mean the number of ASCII symbols in a text but not the size of files containing this text.

The maximal size of the indexed text is about 100 Mb

Text indexation rate is about 1 Mb per min (the average indexation rate 100 Mb)

Time of index opening is not more than 1 min.

Search time is about 1 sec.

It should be noted that the technology being developed is not language dependent and can be adjusted to any language systems. Development of ideas put in searching the similar allows one to solve such problems as search of plagiarism, *rubrication* and *text clusterization*, Internet content filtration and anti-spam system creation.

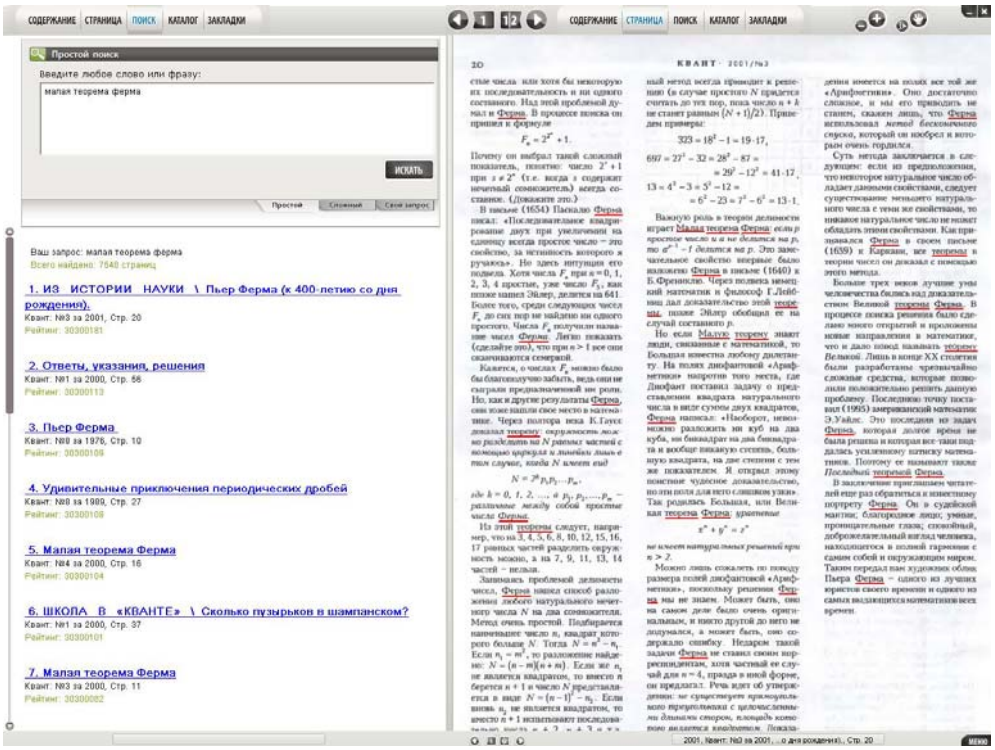


Fig. 2. Search system operation with electronic archive as an example

Bibliography

[Dmitriev etc., 1991] Dmitriev A.S., Panas A.I., Starkov S.O. Storing and recognition information based on stable cycles of 1-D map // Phys. Lett. A.–1991–. V.155, –№8–pp.494-499.

[Dmitriev, 1991] Dmitriev A.S .Record and recognition of information in one-dimensional dynamic systems // РЭ. –1991–V.36– №1–p. 101-108.

[Andreyev etc., 1992] Andreyev Y.V., Dmitriev A.S., Chua L.O., Wu C.W. Associative and random access memory using one-dimensional maps // IJBC–1992–V.3–№3–pp.483-504.

[Andreyev etc., 1997] Andreyev Yu. V., Dmitriev A.S., and Starkov S.O. Information Processing in 1-D Systems with Chaos, *IEEE Transactions on Circuits and Systems*, 1997, vol. 44, No. 1, pp. 21-28.

[Andreyev etc., 1994] Andreyev J.V., Belskij J.L., Dmitriev A.S. Record and restoration of the information with the application of steady cycles of two-dimentional multivariate displays // РЭ–1994–V.39–№1–p.114-123.

[Andreyev etc., 1999] Andreyev Yu. V., Dmitriev A.S., and Ovsyannikov A.V. Information searching system "Forget-Me-Not" based on complex dynamics of nonlinear systems, *Proc. 7th Int. Workshop NDES-99*, 1999. Ronne, Denmark. pp.273-276.

Author‘s Information

Kalafati Yuri Dmitrievich *– Senior Scientist, PhD, Institute or Radio Engineering & Electronics, RAS, Moscow; e-mail: kalafati@controlchaostech.com*

Moiseyev Konstantin Vladimirovich *– Director, Controlling Chaos Technologies, Moscow; e-mail: moiseyev@controlchaostech.com*

Starkov Sergey Olegovich *– Professor, Obninsk State Technical University of Nuclear Power Engineering, Kaluga region, Obninsk; email: Starkov@iate.obninsk.ru*

Shushkova Svetlana Alexandrovna *– Undergraduate, Obninsk State Technical University of Nuclear Power Engineerin, Kaluga region, Obninsk; e-mail: shushkovasvetlana@ramler.ru*

CRITERIA OF LOAN WORDS IDENTIFICATION

Alla Zaboleeva-Zotova, Ilya Prokhorov

***Abstract**: This paper describes the criteria of words relation degree identification based on main adoption methods.*

***Keywords**: Criteria, adoption methods, loaned words.*

***ACM Classification Keywords**: I.6 Simulation and modeling - Model Development*

The development of the modern languages is continuing in the present time. One of the important ways of this development is the loaning of the words. It was being used for different reasons. The basic one is the absence of a suitable word for some concept or object name in the language. But the essential one is the influence of the fashion. For example, the words *корсаж, пальто, буфет, салон, мебель, туалет, бульон* and *котлета* appeared in the Russian language due to the fashion on French language during the reign of the tsar Peter I and later in the end of the 18th – beginning of the 19th centuries.

The identification of the source and the method of word loaning are defined during etymological analysis. This problem has a high dimension even in the case of the direct source identification. The situation is even more complex considering that often a chain of loaning of the word includes several languages. For example, Italian words *купол, кавалер, бензин, коридор* appeared in Russian language by being loaned from French, where they originally appeared from Italian.

In spite of the problems described there has never been any specialized computer system developed for automated etymological analysis.

There certainly are data mining systems that allow to reduce target area of the search, but the major part of the work is being done manually be people.

Considering the automation of the etymological analysis to be the primer goal the authors have developed fuzzy criteria for identification of the loaned words. These criteria are capable to discover words that have been loaned by using any of the following five ways: lexical-word-formative tracing; lexical-word-formative half-tracing; semantic tracing; transcription; transliteration.

Lexical-word-formation is defined in this paper as a literal translation of word parts (prefixes, root, suffixes) with exact imitation of a way of its formation and semantic. The words *кислород* and *водород* are an example that illustrates the usage of this method.

The result of criterion $\mu_f(w_{l_i}, w_{l_j})$ that defines the relationship extent between words $w_{l_i} \in l_i$ and $w_{l_j} \in l_j$ based on lexical-word-formation tracing is computed by the following algorithm:

1. Find all possible translations of the parts of the word w_{l_j} received as a result of the morphological analysis into language l_i;
2. Make fuzzy comparison of translations of the word w_{l_i} parts with the corresponding parts of the word w_{l_i};
3. If a number of translations (combinations of the translated parts of the word) for the parts of the word w_{l_i} forming a word w_{l_i} with average accuracy $\gamma > 60\%$ is found as a result of step 2 then $\mu_f(w_{l_i}, w_{l_j}) = \gamma$, otherwise $\mu_f(w_{l_i}, w_{l_j}) = 0$.

Lexical-word-formative half-tracing is carried out by literal translation of the foreign word parts and adding to them parts from analyzed language. For example, the word *гуманн-остьь* in Russian language has been received by this method from a Latin root *human-us* and Russian suffix «*-остьь*».

The result of criterion $\mu_h(w_{l_i}, w_{l_j})$ that defines the relationship extent between words $w_{l_i} \in l_i$ and $w_{l_j} \in l_j$ based on lexical-word-formation tracing is computed by the following algorithm:

1. Find all possible translations of the parts of the word w_{l_j} received as a result of the morphological analysis into language l_i;
2. Make fuzzy comparison of translations of the word w_{l_i} parts with the corresponding parts of the word w_{l_i};
3. If a number of translations (combinations of the translated parts of the word) for the parts of the word w_{l_i} forming a word w_{l_i} with a minimum accuracy more then 60% is found as a result of step 2 or a maximum accuracy of all found adequacies found as a result of step 2 is less then 60%, then $\mu_h(w_{l_i}, w_{l_j}) = 0$;
4. If only a part of the word w_{l_i} can be formed as a result of comparison taken on step 2 with an accuracy $\gamma > 60\%$ and the remaining part of the word can be formed by using grammars of the morphological analysis of language l_i then $\mu_h(w_{l_i}, w_{l_j}) = \gamma$, otherwise $\mu_h(w_{l_i}, w_{l_j}) = 0$.

Semantic tracing implies assignation of a new semantic meaning to a word from analyzed language under the influence of another language. For example, Russian word *картина* that designated "painting", "spectacle", under the influence of the English language began to be used in the meaning of "movie".

The result of criterion $\mu_s(w_{l_i}, w_{l_j})$ that defines the relationship extent between words $w_{l_i} \in l_i$ and $w_{l_j} \in l_j$ based on semantic tracing is computed by the following algorithm:

1. Enter all homonyms of words w_{l_i} and w_{l_j} into the sets $O_{w_{l_i}}$ and $O_{w_{l_j}}$ accordingly;
2. If $\left|O_{w_{l_i}}\right| = 1$ and/or $\left|O_{w_{l_j}}\right| = 1$, then $\mu_s(w_{l_i}, w_{l_j}) = 0$;
3. If $\left|O_{w_{l_i}}\right| > 1$ and $\left|O_{w_{l_j}}\right| > 1$, then for all $w_i \in O_{w_{l_i}}$:

   3.1 Make fuzzy comparison of the word's w_i synonyms and the synonyms of all words from the set $w_j \in O_{w_{l_j}}$;

   3.2. If there is at least one pair of the synonyms of words w_i and w_j that are congruent with an accuracy $\gamma > 60\%$, then $\mu_s(w_{l_i}, w_{l_j}) = \gamma$;
4. If there are no pairs of synonyms of the words w_i and w_j that are congruent with an accuracy $\gamma > 60\%$ discovered during the step 3, then $\mu_s(w_{l_i}, w_{l_j}) = 0$.

The result of criterion $\mu_\varphi(w_{l_i}, w_{l_j})$ that defines the relationship extent between words $w_{l_i} \in l_i$ and $w_{l_j} \in l_j$ based on transcription is computed by the following method:

If a congruency with an accuracy $\gamma > 60\%$ is a result of comparison of the $\varphi(w_j)$ and $\varphi(w_i)$, where $\varphi()$ is an operation of transcription then $\mu_\varphi(w_{l_i}, w_{l_j}) = \gamma$, otherwise $\mu_\varphi(w_{l_i}, w_{l_j}) = 0$.

The result of criterion $\mu_\tau(w_{l_i}, w_{l_j})$ that defines the relationship extent between words $w_{l_i} \in l_i$ and $w_{l_j} \in l_j$ based on transliteration is computed by the following method:

If a congruency with an accuracy $\gamma > 60\%$ is a result of comparison of the $\tau(w_{l_i}, l_j)$ and w_{l_j}, where $\tau()$ is an operation of transcription then $\mu_\tau(w_{l_i}, w_{l_j}) = \gamma$, otherwise $\mu_\tau(w_{l_i}, w_{l_j}) = \gamma$.

The final relationship extent between words w_{l_i} and w_{l_j} is defined as:

$$\lambda(w_{l_i}, w_{l_j}) = \max(\mu_f(w_{l_i}, w_{l_j}), \mu_h(w_{l_i}, w_{l_j}), \mu_s(w_{l_i}, w_{l_j}), \mu_\varphi(w_{l_i}, w_{l_j}), \mu_\tau(w_{l_i}, w_{l_j}))$$

The described criteria system covers all major ways of words loaning and allows the authors to define the model of the automated etymological analysis.

Authors' Information

Alla Zaboleeva-Zotova *- 400066, CAD department, Volgograd State Technical University, pr. Lenina 28, Volgograd; e-mail: zabzot@vstu.ru*

Ilya Prokhorov *- 400066, CAD department, Volgograd State Technical University, pr. Lenina 28, Volgograd; e-mail: ilya.prokhorov@gmail.com*

КРИТЕРИИ ИНФОРМАТИВНОСТИ И ПРИГОДНОСТИ ПОДМНОЖЕСТВА ПРИЗНАКОВ

Ирина Борисова, Николай Загоруйко, Ольга Кутненко

***Аннотация**: Предлагается вариант решения проблемы Колмогорова о выборе подсистемы признаков, которая была бы не только информативна, но и пригодна для распознавания контрольной выборки. Способ решения состоит в использовании нового критерия информативности признаков в виде функции сходства и различия.*

***Ключевые слова**: Распознавание образов, информативность, пригодность, функция сходства и различия, компактность.*

***ACM Classification Keywords**: Pattern analysis I.5.2.*

Введение

В 1933 году А.Н. Колмогоров опубликовал работу [1], в которой обратил внимание на трудности, связанные с решением проблемы выбора подмножества информативных предикторов при построении регрессионных уравнений для случая, когда количество потенциальных предикторов сравнимо или превышает количество наблюдаемых объектов. Если предикторы зависят друг от друга, то выбор наиболее информативного подмножества из их большого исходного количества представляет собой NP-трудную переборную задачу. Но дело не только в этом. Встречаются задачи, в которых основная часть характеристик не имеет прямого отношения к целевой функции и потому играет роль случайного шума. Чем больше таких характеристик и чем меньше наблюдаемых объектов, тем выше вероятность обнаружения «псевдоинформативного» набора из шумовых предикторов.

В последние годы актуальность проблемы выбора информативного подмножества признаков и оценки его пригодности для решения задач регрессионного анализа и распознавания образов сильно возросла. Стали встречаться реальные задачи распознавания образов, например, в генетике, в которых небольшое число (десятки) объектов обучающей выборки описывается очень большим числом характеристик (десятками тысяч).

Успех в решении этой проблемы зависит от того, как организована **процедура** направленного перебора вариантов, по каким **критериям** оценивается **информативность** и **пригодность** конкурирующих вариантов подсистем признаков. Первая составляющая успеха в решении задачи выбора подсистем признаков претерпела в последние годы заметное развитие. В данной работе рассматриваются вторая и третья часть проблемы Колмогорова, связанные с критериями информативности и неслучайности подсистем-конкурентов.

Вероятность случайного выбора

Если объем обучающей выборки (M) мал, а количество исходных признаков (N) велико, то вероятность того, что в состав информативного подмножества из n<N признаков могут попасть случайные признаки.

Ясно, что эта вероятность будет увеличиваться с ростом размерности выбираемого подпространства n и отношения N/M. Для оценки характера зависимости вероятности случайного результата от параметров N, M и n был проделан машинный эксперимент с таблицами случайных чисел. Количество объектов было равным 75, а размерность таблицы менялся от 10 до 2000. Объекты случайным образом делились на два класса (по 50% объектов в каждом образе). В каждой такой таблице методом AdDel [2] выбирались подсистемы из наиболее информативных признаков. Количество признаков в подсистемах менялось от 1 до 22. На рис. 1 показаны усредненные по 10 экспериментам результаты распознавания обучающей выборки в режиме скользящего экзамена по выбранным признакам.

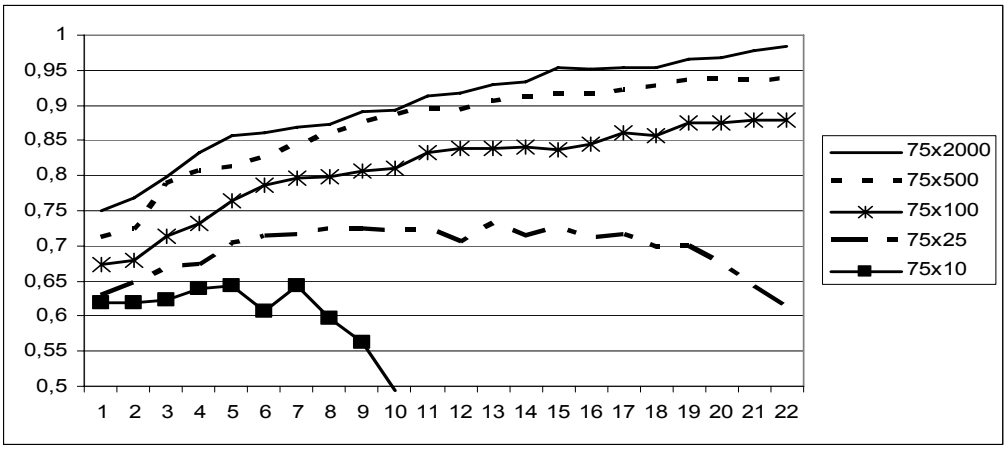


Рис. 1. Вероятность правильного распознавания случайной обучающей выборки для таблиц с параметрами $M = 75$ и N от 2000 до 10 при n от 1 до 22..

Из результатов видно, что при больших N можно найти случайное сочетание n случайных признаков, которые на обучающей выборке покажут свою высокую информативность.

Обратим теперь внимание на следующий вопрос: какой критерий информативности признаков будет защищать нас от случайного выбора наиболее эффективно?

Функция сходства и различия

В описанных экспериментах, как и в большинстве существующих методов, оценкой информативности U подсистем на этапе обучения служило количество правильно распознанных объектов обучающей выборки в режиме скользящего экзамена. При этом решающее правило, по которому контрольный объект z относился к первому образу, было основано на сравнении расстояний r от контрольного объекта z до эталонов первого (r1) и второго (r2) образов. Зная эти расстояния, можно использовать простое правило ближайшего соседа (kNN): если r1<r2, то объект z принадлежит первому образу, и наоборот. Но оказалось, что знанием величин r1 и r2 можно воспользоваться и более эффективно, если ввести следующие функции:

$$F_1 = (r_2 - r_1)/(r_1 + r_2) \text{ и } F_2 = (r_1 - r_2)/(r_1 + r_2)$$

Значения этих функций меняются в пределах от +1 до -1, а их сумма всегда равна 0. Если контрольный объект z совпадает с эталоном первого образа, то r1=0 и F1=1, а F2= -1. Это говорит об абсолютном сходстве объекта z с эталоном первого образа и о максимальном его отличии от эталона второго образа. При расстояниях r1=r2 значения F1=F2=0, что указывает на границу между образами. В точках границы объект в равной степени похож и не похож на эти конкурирующие образы. Функция F хорошо согласуется с механизмами восприятия сходства и различия, которыми пользуется человек, сравнивая некий объект с двумя другими объектами. Мы будем называть F функцией сходства и различия (ФСР). Функция ФСР применима для решения многих задач анализа данных: для автоматической классификации, построения

решающих правил и других. Оказалась она полезной и в качестве критерия информативности признаков. Если, например, объекты двух образов представлены двумя линейно разделимыми группами объектов, то оценка информативности, найденная по критерию U числа правильно распознаваемых объектов, не будет зависеть от расстояния между группами. А среднее значение функции сходства и различия (Fcp) будет зависеть от того, как близко группы находятся от разделяющей границы. Те объекты, которые располагаются в тесном окружении своих объектов и значительно удалены от объектов других образов, имеют более высокое значение функции F, чем периферийные объекты, близкие к другим образам.

Возникла идея сравнить между собой два критерия информативности – число правильно распознанных объектов обучающей выборки (U) и среднее значение функции принадлежности (F_S). Третий критерий вытекает из предложения Фишера оценивать информативность признаковых систем по величине, пропорциональной расстоянию между математическими ожиданиями образов, деленному на сумму их дисперсий: $Q = |\mu_1 - \mu_2| / (\sigma_1 - \sigma_2)$.

При высокой размерности признакового пространства и малом количестве обучающих объектов можно в качестве аналога математического ожидания образа использовать координаты центра тяжести его объектов, а в качестве дисперсий – среднее расстояние между объектами образа.

Эти три критерия – U, F_S и Q - сравнивались в следующем модельном эксперименте. Исходные данные состояли из 200 объектов двух образов (по 100 объектов каждого образа) в 100-мерном пространстве. Признаки генерировались так, чтобы они обладали разной информативностью. В итоге около 30 признаков оказывались в той или иной степени информативными, а остальные признаки генерировались датчиком случайных чисел и были заведомо неинформативными. По этой таблице алгоритмом AdDel выбирались наиболее информативные подсистемы размерности n (от 1 до 22). При этом для обучения случайно выбиралось по 35 объектов каждого образа. На контроль предъявлялись остальные 130 объектов.

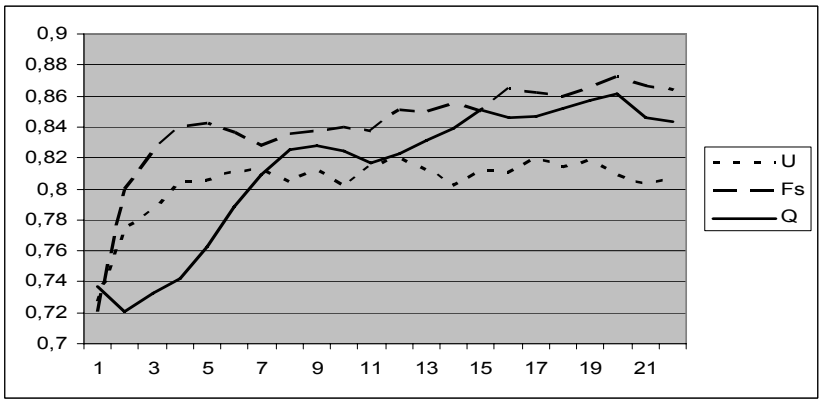


Рис. 2. Результаты выбора подсистем признаков при использовании трех критериев: по числу ошибок (U), по функции принадлежности (F_S) и по критерию Фишера Q.

Надежности распознавания контрольной выборки при использовании критериев U, F_S и Q, усредненные по 10 экспериментам, показаны на рис. 2. Из них видно, что признаки, выбранные по критерию Q, лучше выбранных по критерию ошибок U, но хуже выбранных по функции принадлежности F_S. Это можно объяснить тем, что меры Q и F_S меньше зависят от характеристик отдельных пограничных объектов, чем мера U. В свою очередь, мера Фишера Q ориентирована на разделение нормальных распределений с помощью линейных решающих функций, в то время, как мера F_S адаптируется к особенностям распределения обучающей выборки и соответствует более мощной кусочно-линейной разделяющей границе.

Критерии U и F_S исследовались на устойчивость к помехам. Для этого исходная таблица из предыдущего эксперимента искажалась шумами разной интенсивности и при каждом уровне шума (от 0,05 до 0,3) выбирались наилучшие подсистемы по этим критериям. Результаты представлены на рисунке 3, из которого видно, что критерий F_S более устойчив, чем критерий U. Результаты на контроле показывают высокую степень корреляции критерия Fs с результатами, полученными на обучении. Это свидетельствует о высоких прогностических свойствах критерия Fs.

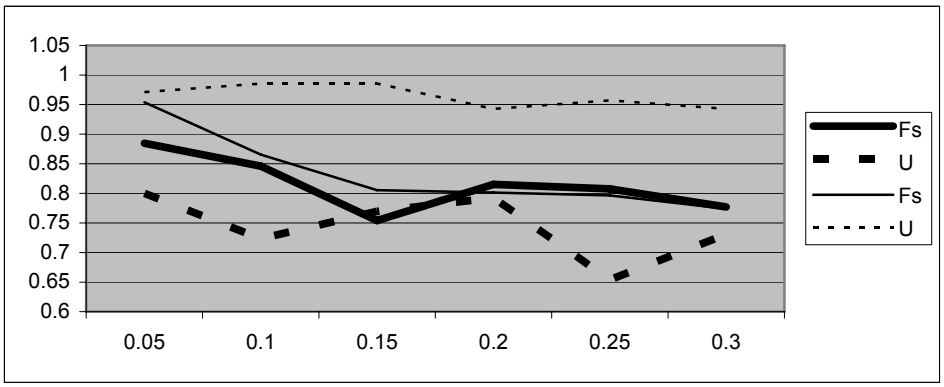


Рис. 3. Результаты обучения и распознавания по критериям U и F_S при разных уровнях шумов. Тонкие линии – обучение, жирные – контроль.

Оценка «пригодности» выбранных подсистем

Для сравнения этих результатов с чисто случайными результатами датчиком случайных чисел с равномерным распределением были сформированы 10 таблиц такого же размера $M = 200$, $N = 100$. Два образа (по 100 объектов) были сформированы методом случайного выбора. По этим чисто шумовым данным для каждой размерности подсистем n выбирались наиболее информативные признаки и определялись значения критерия F_S. Оказалось, что они лежат в «случайном коридоре» с границами от 0,61 до 0,67. Значения F_S для подсистем, найденных по исходной таблице, лежат значительно выше этого коридора и потому могут считаться неслучайными.

Из приведенных результатов можно сформулировать следующую практическую рекомендацию. Определяется значение F_S для наилучшей подсистемы из n^* признаков X, выбранных по обучающей таблице $N \times M$. Затем формируется серия случайных таблиц с такими же значениями N и M, и по ним находятся значения F_S для «лучших» подсистем той же размерности n^*. Если величина F_S для исходной таблицы попадает в пределы значений F_S для случайных таблиц, то можно считать, что выбранные признаки X «псевдоинформативны». Они не пригодны для дальнейшего использования.

По расстоянию между значением критерия F_S подсистемы, выбранной в реальной таблице, и границами «случайного коридора», полученного на наборе случайных таблиц того же размера, можно судить о неслучайности, пригодности выбранных подсистем.

Проверка на реальных данных

Для подтверждения преимуществ критерия F_S перед критерием U на реальных задачах был проведен эксперимент со спектральными данными. Обучающая выборка состояла из двух образов по 25 объектов, выбранных случайно из таблицы реальных спектральных данных двух классов веществ. Из исходного множества 1024 спектральных характеристик формировались два списка из 46 наиболее информативных

«вторичных» признаков в виде не перекрывающихся участков спектра. Один список включал в свой состав признаки, отобранные по критерию U, а второй – по критерию F_S. Затем всем n признакам каждого списка в отдельности предъявлялись для распознавания 200 контрольных объектов (по 100 объектов каждого образа). Надежность распознавания по каждому из 46 наиболее информативных признаков представлена на рис. 4.

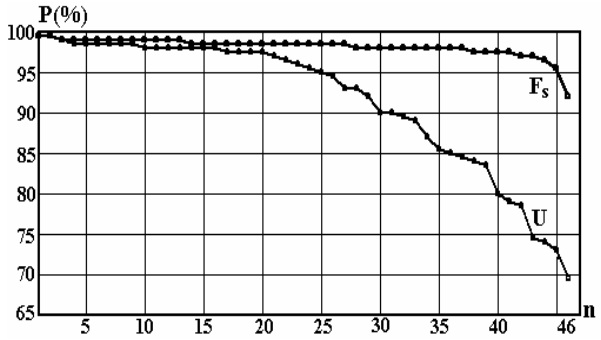


Рис. 4. Надежность P распознавания контрольной выборки по каждому из 46 наиболее информативных признаков, упорядоченных по информативности. Верхняя кривая соответствует выбору по критерию F_S, нижняя – по критерию U.

Эти результаты также подтверждают преимущество выбора признаков по среднему значению функции сходства и различия (F_S) по сравнению с широко распространенным выбором по числу правильно распознанных объектов обучающей выборки (U).

Заключение

Проведенные исследования позволяют сделать следующие выводы:

1. Для оценки информативности признаков или признаковых систем следует использовать не количество правильно распознанных объектов обучающей выборки (U), а среднее значение функции F_S сходства объектов обучающей выборки с эталонами своих образов.

2. Значения меры F_S, получаемые на обучающей таблице и на серии случайных таблиц того же размера, позволяют получить качественную оценку пригодности выбранного подпространства признаков.

3. Значение функции F сходства контрольного объекта с эталоном того или иного образа дает возможность сопроводить результат распознавания оценкой правильности этого результата.

Библиография

1. Колмогоров А.Н. К вопросу о пригодности найденных статистическим путем формул прогноза. - Заводская лаборатория. 1933. №1. С. 164-167.

2. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: Изд. ИМ СО РАН, 1999.

Информация об авторах

Ирина Борисова – *Институт Математики СО РАН, пр. Коптюга, дом 4, Новосибирск, 630090, Россия; e-mail: biamia@mail.ru*

Николай Загоруйко *- Институт Математики СО РАН, пр. Коптюга, дом 4, Новосибирск, 630090, Россия; e-mail: zag@math.nsc.ru*

Ольга Кутненко *- Институт Математики СО РАН, пр. Коптюга, дом 4, Новосибирск, 630090, Россия; e-mail: olga@math.nsc.ru*

CONDITIONS OF EFFECTIVENESS OF PATTERN RECOGNITION PROBLEM SOLUTION USING LOGICAL LEVEL DESCRIPTIONS OF CLASSES

Adil Timofeev, Tatiana Kosovskaya

Abstract: *Earlier the authors have suggested a logical level description of classes which allows to reduce a solution of various pattern recognition problems to solution of a sequence of one-type problems with the less dimension. Here conditions of the effectiveness of the use of such a level descriptions are proposed.*

Keywords: *compound images, logical description of classes, effectiveness.*

Introduction

Various pattern recognition problems wich may be described in the terms of predicates (which characterize the whole object or its parts) were reduced in [1] to the proof of deducibility of propositional and predicat calculus formulas from a set of atomic formulas.

Upper bounds of the number of steps of an algorithm solving pattern recognition problems with logical description were proved in [3]. For example, such an upper bound for an algorithm solving the problem of analysis of a compound object is polynomial but the degree of such a polynomial depends of the number of objective variables included to the class description. As a rule such a number is rather large.

A level description of classes offered in [2] allows to reduce the solution of various pattern recognition problems to the solution of a sequence of one-type problems with the less dimension. Here conditions of decreasing of the number of steps of algorithm solving the described pattern recognition problems with the use of many-level description are proposed.

Setting of a problem of compound objects logical recognition

Let Ω be a set of finite sets $\omega = \{\omega_1, \ldots, \omega_l\}$. The set ω will be called a recognizable object. Let $p_1, \ldots, p_n$ be a collection of predicates which characterize an object (global indication) or describe properties or relations between elements of ω (local indication). The set Ω is a union of K (may be intersected) classes Ω_k.

Logical description $S(\omega)$ of an object ω is a set of all true formulas in the form $p_i(x)$ or its negation written for all parts x of the object ω.

Logical description of a class (DC) Ω_k is such a formula $A_k(x)$ that $A_k(x)$ contains as an atomic only formulas of the form $p_i(y)$ where y is a subset of x; $A_k(x)$ has no quantifiers; if for some ordering ω' of the object ω the formula $A_k(\omega)$ is true then $\omega \in \Omega_k$

These descriptions may be used for solving the following problems.

Identification problem. To check weather object ω or its part belongs to the class Ω_k.

This problem was reduced in[1] to the proof of deducibility of the formula $\exists y\ (y \subset \omega\ \&\ A_k(y))$ from the description $S(\omega)$.

Classification problem. To find all such numbers k that $\omega \in \Omega_k$.

This problem was reduced in[1] to the proof of deducibility of disjunction of formulas $A_k(\omega')$ (for some ordering ω' of the object ω) from the description $S(\omega)$ and pointing out all such numbers k for which the corresponding disjunct is true for ω.

Problem of analysis of a compound object. To find and classificate all parts x of the object ω.

This problem was reduced in[1] to the proof of deducibility of disjunction of formulas $\exists y\ (y \subset \omega\ \&\ A_k(y))$ from the description $S(\omega)$ and pointing out all parts of ω wich mai be classified.

2. Level logical description of classes

Objects the structure of which allows to extract more simple fragments and to describe these objects in the terms of properties and relations between such fragments are regarded. In particular it may be done be means of selecting «frequently» appeared subformulas of formulas $A_k(x)$ with «small complexity». A system of equivalences in the form $p_j^1(x_j^1) \Leftrightarrow P_j^1(y_j^1)$ (where x_j^1 – new first-level variables, p_j^1 – new first-level predicates, $P_j^1(y_j^1)$ – subformulas of formulas $A_k(x)$) is written. The result of substitution of $p_j^1(x_j^1)$ instead of $P_j^1(y_j^1)$ into $A_k(x)$ is denoted by $A_k^1(x^1)$.

Such a procedure may be repeated with $A_k^1(x^1)$ but not later than $A_k^l(x^1)$ contains at least two subformulas in the same form.

3. Conditions of effectiveness of level description with the use of global indications

Let $p_1, \dots, p_n$ be global indications (i.e. they are boolean variables). Then class descriptions are disjunctive normal forms (DNF) and any subformula of formulas $A, \dots, A_K$ which appears at least two times is a simple conjunction.

Definition. Atom is variable or its negation.

Definition. Simple conjunctions $B_1, \dots, B_m$ are called disjoint if there not exists such an atom that is included simultaneously into two different conjunctions.

Notifications.

a – a number of occurrences of boolean variables in formulas in DNF $A_1, \dots, A_K$,

$P_1^1, \dots, P_{n1}^1$ – subformulas of $A_1, \dots, A_K$,

N_j^1 – a number of occurrences of subformula P_j^1,in $A_1, \dots, A_K$,

v_j^1 – a number of occurrences of boolean variables in P_j^1,

$A_1^1, \dots, A_K^1$ – the result of substitutions of atomic formulas p_j^1 instead of P_j^1 into $A_1, \dots, A_K$,

Theorem 1. *If formulas $P_1^1, \dots, P_{n1}^1$ are disjoint then for the equality $a^1 = d\,a$ (for some $0 < d < 1$) it is necessary and sufficient*

$$\Sigma_{j=1}^{n1} (v_j^1 - 1) N_j^1 = (1-d)\, a . \quad (1)$$

Corollary 1.1. *If formulas $P_1^1, \dots, P_{n1}^1$ are disjoint then for decreasing the number of occurrences of boolean variables in formulas $A_1^1, \dots, A_K^1$ in comparison with the number of occurrences of boolean variables in formulas $A_1, \dots, A_K$ it is necessary and sufficient*

$$\Sigma_{j=1}^{n1} (v_j^1 - 1) N_j^1 > a . \quad (2)$$

Corollary 1.2. *If formulas $P_1^1, \dots, P_{n1}^1$ are disjoint and $N_j > N$ for some N then for decreasing the number of occurrences of boolean variables in formulas $A_1^1, \dots, A_K^1$ in comparison with the number of occurrences of boolean variables in formulas $A_1, \dots, A_K$ it is necessary and sufficient*

$$\Sigma_{j=1}^{n1} (v_j^1 - 1) \geq a / N . \quad (3)$$

The next theorem gives a necessary condition for not disjoint formulas $P_1^1, \dots, P_{n1}^1$.

Theorem 2. *For the equality $a^1 = d\,a$ (for some $0 < d < 1$) it is necessary and sufficient*

$$\Sigma_{j=1}^{n1} (v_j^1 - 1) N_j^1 \geq (1-d)\, a . \quad (4)$$

If $p_1, \dots, p_n$ are boolean variables then both the identification problem and the classification problem may be solved with the use of resolution method or sequent propositional calculus the number of rule applications of which is not more then the number of occurrences of boolean variables in formula A_k (in formulas $A_1, \dots, A_K$ for classification problem) [3]. Note that the upper bound of number of steps needed for calculation of p_j^1 equals to such a bound for classification problem if instead of A_k we take P_j^1.

Theorem 3. *If $a^1 = d\,a$ then for decreasing of number of rule application steps while using the 2-level description it is sufficient*

$$\Sigma_{j=1}^{n1} v_j^1 \leq (1-d)\, a. \quad (5)$$

4. Examples of two-level descriptions

Illustrate an application of the received conditions with a model example of 2-level description.

Let the set of recognizable objects is divided into 3 classes and objects may be described by means of 5 boolean variables *x, y, z, u, v*. Classes descriptions which allow to identify and classify an object have the form

$A_1 = \sim x \,\&\, \sim y \;\; V \;\; x \,\&\, \sim y \,\&\, z \;\; V \;\; x \,\&\, y \,\&\, z \,\&\, \sim v$

$A_2 = \sim x \,\&\, y \,\&\, \sim z \,\&\, \sim u \;\; V \;\; \sim x \,\&\, y \,\&\, u \,\&\, \sim v \;\; V \;\; x \,\&\, \sim z \,\&\, \sim u \;\; V \;\; x \,\&\, z \,\&\, u \,\&\, \sim v$

$A_3 = \sim x \,\&\, y \,\&\, z \,\&\, \sim u \;\; V \;\; \sim x \,\&\, y \,\&\, u \,\&\, v \;\; V \;\; x \,\&\, \sim z \,\&\, u \,\&\, v \;\; V \;\; x \,\&\, y \,\&\, z \,\&\, v$

The number of occurrences of boolean variables in formulas A_1, A_2, A_3 is a=40.

Example 1. Let the following subformulas are extracted.

$P_1^1 = x \,\&\, y \,\&\, z$

$P_2^1 = \sim x \,\&\, y \,\&\, u$

$P_3^1 = \sim x \,\&\, y \,\&\, \sim u$

$P_4^1 = x \,\&\, z$

$P_5^1 = y \,\&\, z$

The number of occurrences of boolean variables in these subformulas is ν_1^1=3, ν_2^1=3, ν_3^1=3, ν_4^1=2, ν_5^1=2. The number of occurrences of each of these subformulas is N_j^1=2.

The formulas are not disjoint and we can use the condition (4) $\Sigma_{j=1}^{n1} (\nu_j^1 - 1) N_j^1 \geq (1-d)\, a$. In this example $\Sigma_{j=1}^{n1} (\nu_j^1 - 1)\, N_j^1 = 16$. Hence for decreasing the length of description it is necessary $16 \leq 40$ (1-d), i.e. $d \geq 0.4$.

In fact

$A_1^1 = \sim x \,\&\, \sim y \;\; V \;\; \sim y \,\&\, p_4^1 \;\; V \;\; \sim v \,\&\, p_1^1 \,\&\, p_4^1 \,\&\, p_5^1$

$A_2^1 = \sim z \,\&\, p_3^1 \;\; V \;\; \sim v \,\&\, p_2^1 \;\; V \;\; x \,\&\, \sim z \,\&\, \sim u \;\; V \;\; u \,\&\, \sim v \,\&\, p_4^1$

$A_3^1 = p_3^1 \,\&\, p_5^1 \;\; V \;\; v \,\&\, p_2^1 \;\; V \;\; x \,\&\, \sim z \,\&\, u \,\&\, v \;\; V \;\; v \,\&\, p_1^1 \,\&\, p_4^1$

The number of occurrences of boolean variables in A_1^1, A_2^1, A_3^1 is a^1=29. As $a^1 = d\, a$ we have d= 29/40 = 0.725.

Verify, weather we may guarantee that such a 2-level description provides a decreasing of an upper bound of number of steps of a solution of classification problem, i.e. weather the condition (5) $\Sigma_{j=1}^{n1} \nu_j^1 \leq (1-d)\, a$ is fulfilled. In this example $\Sigma_{j=1}^{5} \nu_j^1$ =13, $(1-d)\, a$ = 11. Hence the condition (5) is not fulfilled and we can not guarantee a decreasing of an upper bound of number of steps of a solution of classification problem.

Example 1. Let the following subformulas are extracted.

$P_1^1 = \sim x \,\&\, y$

$P_2^1 = x \,\&\, z$

$P_3^1 = y \,\&\, z$

The number of occurrences of boolean variables in these subformulas is ν_1^1=2, ν_2^1=2, ν_3^1=2. The number of occurrences of each of these subformulas is N_1^1=4, N_2^1=4, N_3^1=3.

The formulas are not disjoint and we can use the condition (4) $\Sigma_{j=1}^{n1} (\nu_j^1 - 1) N_j^1 \geq (1-d)\, a$. In this example $\Sigma_{j=1}^{n1} (\nu_j^1 - 1)\, N_j^1 = 11$. Hence for decreasing the length of description it is necessary $11 \leq 40$ (1-d), i.e. $d \geq 0.725$.

In fact

$A_1^1 = \sim x \,\&\, \sim y \;\; V \;\; \sim y \,\&\, p_2 \;\; V \;\; \sim v \,\&\, p_2^1 \,\&\, p_3^1$

$A_2^1 = \sim z \,\&\, \sim u \,\&\, p_1^1 \;\; V \;\; u \,\&\, \sim v \,\&\, p_1^1 \;\; V \;\; x \,\&\, \sim z \,\&\, \sim u \;\; V \;\; u \,\&\, \sim v \,\&\, p_2^1$

$A_3^1 = \sim u \,\&\, p_1^1 \,\&\, p_3^1 \;\; V \;\; u \,\&\, v \,\&\, p_1^1 \;\; V \;\; x \,\&\, \sim z \,\&\, u \,\&\, v \;\; V \;\; v \,\&\, p_2^1 \,\&\, p_3^1$

The number of occurrences of boolean variables in A_1^1, A_2^1, A_3^1 is a^1=32. As $a^1 = d\, a$ we have d= 32/40 = 0.8.

Verify, weather we may guarantee that such a 2-level description provides a decreasing of an upper bound of number of steps of a solution of classification problem, i.e. weather the condition (5) $\Sigma_{j=1}^{n1} \nu_j^1 \leq (1-d)\, a$ is fulfilled. In this example $\Sigma_{j=1}^{3} \nu_j^1$ = 6, $(1-d)a$= 8. Hence the condition (5) is fulfilled and we can guarantee a decreasing of an upper bound of number of steps of a solution of classification problem.

Conditions of effectiveness of level description with the use of local indications

Let $p_1, \dots, p_n$ characterize properties and relations of a recognizable object elements. In such a case it was proved in [3] that the number of steps of an algorithm solving identification problem is bounded by the number of arrangement of m_k from t : A_t^{mk}. For classification problem and problem of analysis of a compound object such a bound is $\Sigma_{k=1}^{K} A_t^{mk}$. (Here m_k – the number of objective variables in the description of the k-th class.

Consequently the number of steps of an algorithm solving these problems is an exponent of the number of objective variables in the description of classes (and a polynomial of a high degree for any particular description). Moreover, if it is possible to construct an algorithm which in a polynomial (over the length of classes descriptions) number of steps solves such problems then one of the most difficult problems of XXI century **P=NP** will be solved. However with the use of level logical descriptions it is possible to decrease an exponent in the upper bound of the number of steps of such an algorithm.

Notifications.

$m_1, \dots, m_K$ – number of objective variables in formulas $A_1, \dots, A_K$,

r – a number which is greater than number of objective variables in every formula $P_1^1, \dots, P_{n1}^1$,

x_k – the string of variables of the formula A_k,

x_j^1 – new variables of the 1st level defined be equivalences $p_j^1(x_j^1) \Leftrightarrow P_j^1(y_j^1)$,

$s1$ – the number of variables occurred in $A_1, \dots, A_K$ but not occurred in $P_1^1, \dots, P_{n1}^1$.

Theorem 4. *Checking weather formulas $A_1, \dots, A_K$ are true on the set $\omega = \{\omega_1, \dots, \omega_t\}$ is equivalent to checking equivalences $p_j^1(x_j^1) \Leftrightarrow P_j^1(y_j^1)$ and weather formulas $A_1^1, \dots, A_K^1$ are true on the same set.*

For decreasing the number of steps of an algorithm solving the problem of analysis of compound object it is sufficient

$$n_1 t^r + t^{s1+n1} < t^m. \qquad (6)$$

Conclusion

Hence level logical description of classes of objects is described. In the frameworks of such an approach the conditions of decreasing of upper bounds of number of steps of an algorithm solving various pattern recognition problems including recognition of compound objects (compound images and scenes, complex signals and so on) are done.

Bibliography

[1]. Timofeev A.V., Kosovskaya T.M. Logic-Axiomatical Method of Knowledge Representation and Recognition for Robots. In: Computers in Industry 15 (1990). Proc. Intelligent Manufactoring Systems – IMS'89. Elsevier Science Publishers B.V. 1990.

[2]. Kosovskaya T.M. Level descriptions of classes for providing a solution in pattern recognition problems. In: Proc. III International Conf. “Discrete models in the theory of control systems”. Moscow, Dialog-MSU, 1998. (in Russian)

[3]. Kosovskaya T.M. Proofs of the number of step bounds for solving of some pattern recognition problems with logical description. In: Herald of St.Petersburg University, Series 1 (Mathematics), 2007 (?). (be published, in Russian)

Authors' Information

Timofeev Adil V. *– Dr.Sc., Professor, Head of the Laboratory of Information Technologies in Control and Robotics of Saint-Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, tav@iias.spb.su*

Kosovskaya Tatiana M. *– Doctor's Degree Student, Laboratory of Information Technologies in Control and Robotics of Saint-Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, Russia; Ph.D., Assistant Professor, Department of Mathematics and Mechanics, Saint-Petersburg State University, Russia, e-mail: kosov@nk1022.spb.edu*

BAYESIAN MODEL OF RECOGNITION ON A SYSTEM OF EVENTS

Vladimir Berikov[1]

***Abstract**: We consider a problem of pattern recognition with use of logical decision functions class. To define an optimal complexity of the class, we suggest Bayesian model of recognition on a system of events. This model takes into account empirical data as well as expert knowledge on problem domain. Some properties of the model are considered.*

***Keywords**: logical decision function, generalization ability, Bayesian modeling*

***ACM Classification Keywords**: I.5.2 Pattern recognition: classifier design and evaluation*

Introduction

At present time, in many hard-to-formalize areas of investigations (biology, archeology, medicine etc) one should solve Data Mining tasks characterizing by the following peculiarities:

- lack of knowledge about the objects under investigation, which makes it difficult to formulate the mathematical model of the objects;
- large number of heterogeneous (either quantitative or qualitative) features under comparatively small sample size;
- nonlinear dependencies between features;
- presence of unknown values of features or errors in their measurements;
- need to forecast rare events connected with large losses at their wrong prediction;
- desire to present the results of the analysis in the form understandable by a specialist in the applied area.

Ones of the most promising methods for solving the problems of such type are the methods based on logical decision functions class. The convenient form of logical decision function is decision tree. It is known that the complexity of decision function class (VC-dimension, capacity, maximal number of logical rules in case of logical functions with fixed type of predicate) is an important factor influencing the generalization ability of a prediction method. To achieve the best quality, one has to obtain some compromise between the complexity and the accuracy of the decision on learning sample.

In a large number of applied problems one can use, along with empirical data (learning sample), different types of expert knowledge (which may have the form of some estimates; express restrictions on the class of distributions or decision functions; define preference rules for decision making etc) having no connections with "hard" definition of probability distribution model. Therefore at the choice of optimal complexity of the class one should consider empirical data as well as expert knowledge. To do this, one can apply the Bayesian learning theory. This approach is based on the idea of using a priory knowledge on the problem, which allows in particularly to assign for every possible distribution ("strategy" or "state" of nature) some weight. This weight reflects the intuitive expert's believe that the unknown true distribution coincides with the considered one.

In this paper, we suggest the Bayesian model of recognition on a system of events. This model allows theoretically to choose the optimal complexity of logical decision functions class taking into account both empirical data and expert knowledge.

Logical decision functions in forecasting problems

In the problem of forecasting one should predict the value of some dependent feature Y for arbitrary object from general collection of objects Γ, there each object $a \in \Gamma$ can be described by features $X_1,\ldots,X_n$. Here the prediction

[1] This work is supported by RFBR projects 07-01-00331-a and 07-01-00393-a.

is carried out on the basis of the analysis of learning sample. It is supposed that the features can be heterogeneous, i.e. some part of them can be of qualitative nature, and the another part can be of quantitative nature. For qualitative *Y* we have the problem of pattern recognition, and for quantitative *Y* we have the regression analysis problem.

As a rule, for the solution of the problem one should apply some class of decision functions, in which the optimal function (by some quality criterion) should be found. The class of logical decision functions is defined on the set of divisions of feature space on a finite number of sub-regions describing by conjunction of predicates of simple form. The number of sub-regions defines the complexity of logical function. The convenient form of logical decision function is decision tree. More thorough definition and properties of logical decision functions class one can find in [1].

Bayesian model

The Bayesian model of recognition on a system of events is introduced by formulation some propositions, the sense of which consists in abstracting from local metric properties of feature space and local characteristics of learning method:

- transition from points of feature space to the “events”, where under “event” we understand taking by features the values from some sub-region of space;
- forming the system of the events taking into account relations between them (neighborhood or common ancestor);
- using the notion of learning method (mapping from the set of all possible samples into the set of decision functions) and considering different ways of formalization the expert knowledge about forecasting problem (not demanding “hard” definition of probability distribution model).

One can suggest two base methods of forming the system of events.

1. Let us form the initial partition of feature space on a certain sufficiently large number of sub-regions. Each sub-region is the Cartesian product of subsets of feature values (for quantitative feature, it must be partitioned beforehand on intervals whose lengths are defined coming from known inaccuracy of measurements). Let us consider certain partition tree. We shall consider the initial set of sub-regions, or certain “integration” of them, received as a result of merging some of the sub-regions in accordance with given tree structure.

2. Let us use another way of initial tree formation. Divide randomly training sample in two parts. The first part serves for building a decision tree by some algorithm, for instance, by means of consequent branching. The parameters of the algorithm should be chosen in such a way to ensure the minimum empirical risk of the decision. The received tree corresponds to the system of events (leaves). The second part of the sample is used for pruning (simplification) the tree. The described method is attractive in the sense that the initial tree does not have “empty” leaves (i.e. such ones, to which no one object belongs). So the number of leaves is relatively small.

In [2], the Bayesian model of recognition on finite set of events was introduced. Let us describe shortly the model. Let X be discrete random variable taking non-ordered values (cells, events) from set D_X={1,2,…,M} and let *Y* be another discrete random variable taking values from D_Y={1,2,…,K}. Let $p_j^{(i)}$ be the probability of joint event *“X=j,Y=i”*, $j=1,...,M$, $i=1,...,K$. Let some *decision functions class* Φ be defined; $\Phi=\{f\}$, f: $D_X \to D_Y$. The value *M* will be called *the complexity* of the class. It is supposed that *loss function* $L_{r,l}$ defines losses for the situation when one makes the decision $Y=r$ but the true value of Y is l. For every decision function $f \in \Phi$ we can calculate the expected losses (*risk*) of forecasting for arbitrary observation: $R(\theta)=\sum_{i,j} L_{f(j),i} p_j^{(i)}$. In p*attern recognition* problems, zero-one (“0,1”) loss function is usually applied (in this case, the risk coincides with misclassification probability). In *regression analysis* problems, the quadratic loss function $L_{r,l}=(r-l)^2$ can be used. The decision function has to be selected from Φ with use of some *method* μ on the basis of random sample over *X* and Y (learning sample). Let N be sample size, $n_j^{(i)}$ be a frequency of falling of the observations

from i-th class into j-th cell: $s = (n_1^{(1)}, n_1^{(2)}, ..., n_1^{(K)}, n_2^{(1)}, ..., n_M^{(K)})$. Let us consider the family of models of multinomial distributions with set of parameters $\mathbf{\Theta} = \{\theta\}$. We use the Bayesian approach: suppose that random vector Θ (the *"state of nature"*) having known priory distribution $p(\theta)$ is defined on the set of parameters. We shall suppose that Θ is subject to Dirichlet distribution: $p(\theta) = \frac{1}{Z}\prod_{i,j}(p_j^{(i)})^{d-1}$, where d> 0 are some given real value expressing a priori knowledge about distribution of Θ, i=1,...,K, j=1,...,M, Z is normalizing constant. If d=1, then $p(\theta)$ =const (this is the case of the uniform a priori distribution, then there is no information about preferences between the states of nature). For the fixed vector of parameters Θ, the probability of error for the Bayesian classifier f_B is: $P_{f_B}(\Theta) = \sum_j \min\{p_j^{(1)}, p_j^{(2)}\}$ (for K=2). In [2], the expected probability of error $EP_{f_B}(\Theta)$ was found, where the averaging is done over all random vectors Θ with distribution density $p(\theta)$:

Proposition 1. $EP_{f_B}(\Theta) = I_{0,5}(d+1, d)$, where $I_x(p,q)$ is beta distribution function.

Parameter d can be used for the definition of a priori distribution on recognition tasks: when this parameter decreases, the density of a priori distribution is changed so that classes are less intersected in average.

The expected risk R_μ for the method μ is defined as mathematical expectation $E_{S,\Theta}R_{\mu(S)}(\Theta)$.

Proposition 2. Let 0-1 loss function be given, the method μ^* be the method that minimizes empirical error and the Dirichlet parameter d be fixed. Then the expected misclassification probability is

$$P_{\mu^*} = \frac{N!M}{(2Md)_{(N+1)}} \sum_{\substack{i,j,l\geq 0: \\ i+j+l=N}} \frac{(2Md-2d)_{(l)}(d)_{(l)}(d)_{(j)}}{l!i!\,j!}(d+\min\{i,j\}),$$

where $a_{(n)}$ denotes the product $a(a+1)\ldots(a+n-1)$.

Let us consider the dependency of the expected misclassification probability from the complexity of decision functions class. Let in accordance with the structure of the given system of events a sequence of classes be formed with increasing complexity M=1,2,...,M_{max}. When increasing the complexity of the class it is naturally to suppose that the expected probability of error $EP_{f_B}(\Theta)$ for optimum Bayesian decision function (expressing the degree of the intersection between patterns) also has to be changed. Under M=1 this value is maximum since in this case the optimum Bayesian decision is based on a priori probabilities only, but under the further increase of M the value $EP_{f_B}(\Theta)$, as a rule, should monotonously decrease (converges to zero when the set of events is formed by partition of the real space).

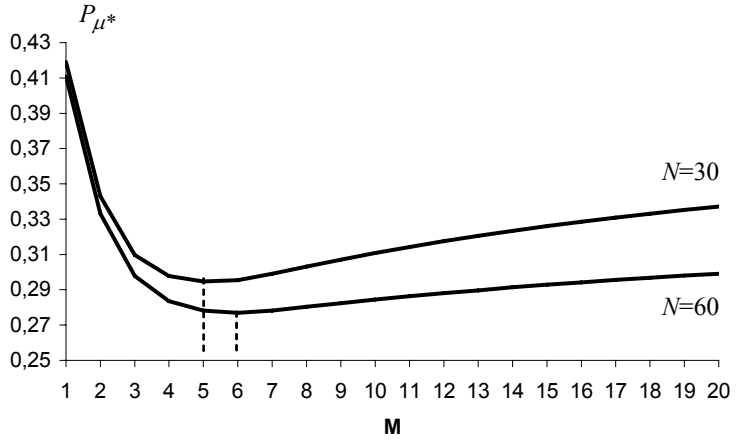


Fig.1

Herewith under small values of M the complication of model (transition from M to M+1) should cause the noticeable reduction of $EP_{f_B}(\Theta)$, but under greater values of M the effect of such complication is less expressing. For each M we shall denote the expected optimum probability of error through $EP_B(M)$, which corresponds to the value of the Dirichlet parameter d_M. The choice of concrete values $d_1, d_2, \ldots, d_{M_{max}}$ (or corresponding values $EP_B(1)$, $EP_B(2)$,..., $EP_B(M_{max})$) should be done on the basis of expert knowledge.

It is possible to suggest, for instance, the following method. Let us define the model of dependency for $EP_B(M)$ from M (power or exponential), as well as the extreme values $EP_B(1)$ and $EP_B(M_{max})$. Then in accordance with proposition 1 the values $d_1,d_2,\ldots,d_{M_{max}}$ should be found. Finally, with use of proposition 2 the expected misclassification probabilities are to be calculated for different values of M. Fig. 1 exemplifies the dependency of the expected misclassification probability from the complexity of class. One can see that between the extreme values of M the best value, depending from sample size, exists for which the expected misclassification probability is minimum. The model has the form: $EP_B(M)=(EP_B(1)-EP_B(M_{max}))e^{-0.75(M-1)}+ EP_B(M_{max}))$, $M=2,3,\ldots, M_{max}-1$, $M_{max}=20$, $EP_B(1)=0.4$, $EP_B(M_{max})=0.25$.

Conclusion

In this paper, we briefly described the new approach towards the development and study the methods based on logical decision functions in hard-to-formalize areas of investigations. This approach is founded on Bayesian model of recognition on a system of events. Unlike other approaches, the suggested one possesses a number of advantages: takes into account expert knowledge, allows to considerate the specifics of learning method, does not require postulating distribution model, allows to conduct theoretical research under arbitrary volume of sample. The quality of the model was investigated: the dependencies between the expected risk, learning sample size and complexity of decision functions class were obtained. This allowed to find the optimum complexity of the class depending on sample size and expert knowledge.

Bibliography

[1] Lbov, G.S., Berikov V.B. Stability of decision functions in problems of pattern recognition and analysis of heterogeneous information. Sobolev institute of mathematics, Novosibirsk, 2005. (in Russian)

[2] Berikov V.B. Recognition on Finite Set of Events: Bayesian Analysis of Generalization Ability and Classification Tree Pruning. Int. J. Information Theories & Applications. 2006, Vol.13, N. 3, p. 285-290.

Authors' Information

Vladimir B. Berikov – *Sobolev Institute of Mathematics SB RAS, Koptyug pr.4, Novosibirsk, Novosibirsk State University, Russia, 630090; e-mail: berikov@math.nsc.ru*

ПРОЦЕДУРЫ ЛОКАЛИЗАЦИИ ВЕКТОРА ВЕСОВЫХ КОЭФФИЦИЕНТОВ ДЛЯ НЕЧЕТКИХ МОДЕЛЕЙ ВЫБОРА

Елена Присяжнюк

Аннотация: *Рассматриваются процедуры локализации вектора весовых коэффициентов, основанные на представлении функции полезности аддитивной сверткой, адаптированные к нечеткой модели выбора*

Ключевые слова: *нечеткие множества, коэффициенты важности.*

Вступление

Разного рода неопределенности в той или иной степени присущи практически любой ситуации принятия решений, в которой используется экспертная информация. Результаты исследований [Ларичев, 2002; Борисов, 1989] показывают, что особую сложность вызывает необходимость оценивать числовые значения объектов (вариантов, критериев) или дать числовую оценку на шкале отношений между ними. Известно, что использование словесных определений позволяют более надежно выявлять предпочтения

ЛПР. Такой подход кажется более оправданным, поскольку в преобладающем числе случаев достаточна приближенная характеристика набора данных и большинство задач, где необходимое применение экспертной информации, не требуют высокой точности.

Оценки экспертов в нечетких моделях выбора описываются функциями принадлежности нечеткому множеству. Сами функции принадлежности могут интерпретироваться различным образом: как "субъективная вероятность", степень уверенности эксперта в принадлежности объекта к понятию, описываемому нечетким множеством, возможностью его интерпретации этим понятием и так далее.

Выбор характеризуется отношением предпочтения R, суть которого в нечетких моделях состоит в том, что для каждых двух объектов он может указать:

- факт предпочтения объекта α^1 над α^2 . Функция $\mu_R(\alpha^1, \alpha^2)$ в этом случае содержательно интерпретируется как степень уверенности эксперта в том, что α^1 не менее предпочтительный чем α^2 . Причем степени уверенности могут описываться как численно, так и вербально, например лингвистическими переменными «степень уверенности»={очень низкая, низкая, средняя, высокая, очень высокая}. Лингвистическое отношение предпочтения этого типа отвечает ситуациям принятия решений, когда ЛПР сомневается в наличии предпочтений в отношениях тех или иных объектов и поэтому затрудняется их выразить лишь в терминах «да»(определенно доминирует) или «нет» (определенно доминируется);
- силу предпочтения объекта α^1 над α^2. Нечеткое отношение R в данном случае отображает понятие силы предпочтения; при этом в самом факте предпочтения ЛПР уверенно (и в этом смысле его предпочтения являются четкими). Здесь $\mu_R(\alpha^1, \alpha^2)$ может быть интерпретирована как степень, с которой α^1 определенно лучше (предпочтительней) чем α^2.

Проблема оценки объектов в рамках теории полезности сводится к проблеме аксиоматического обоснования и построения его функции полезности. Классические методы, используемые для определения функции полезности, представляющей бинарное отношение предпочтения R ($U(a^1) \geq U(a^2) \Leftrightarrow a^1 R a^2$ для $\forall a^1, a^2 \in A$), в общем случае являются достаточно «жесткими». Основанием для их применения, в частности, служат достаточные условия ее существования, которые задаются, например, теоремой Дебре [Пономаренко, 1994]: отношение предпочтения должно быть полным, рефлексивным, транзитивным и непрерывным), множество решений – связанным. Если условия теоремы Дебре не выполняются и функция полезности, которая представляла бы отношение R, не существует, применение классических методов затруднено.

Предлагается процедура формализации проблемы путем замены нечеткой «векторной оценки полезности» аддитивной сверткой.

Постановка задачи

Пусть A – обычное (четко описанное) множество объектов α^j, $j \in J$, где J - множество индексов объектов. Каждый из объектов $a^j \in A$, $j \in J$, характеризуется набором параметров $a^j = (a^j{}_1, ..., a^j{}_i, ..., a^j{}_n)$. Множество индексов параметров объектов обозначим I, $I = \{1, ..., n\}$. Каждому объекту, $a^j \in A$, $j \in J$, ставится в соответствие его векторная оценка в пространстве параметров объектов Ω^n.

В дальнейшем будем рассматривать не само множество значений параметров объектов $a^j \in A$, $j \in J$, а соответствующее ему множество $\omega(a^j{}_i)$, $i \in I$, $j \in J$, где ω некоторое монотонное преобразование, которое определяет степень отклонений от оптимальных значений для каждого параметра $a^j{}_i$, $i \in I$, $j \in J$, и преобразует все значения параметров объектов к безразмерному виду в интервале [0,1].

Пусть эксперт последовательно задает свои предпочтения на множестве A в виде нечеткого бинарного отношения предпочтения R.

Предлагается следующий подход к решению задачи: предполагается, что при оценке объекта эксперт (сознательно или неосознанно) имеет в виду его векторную оценку. Если рассматривать «векторную» функцию полезности в виде нечеткой аддитивной свертки, то задача сводится к уточнению весовых коэффициентов свертки (1)-(2):

$$\sum_{i\in I}\rho_i\omega(a_i^1) < \sum_{i\in I}\rho_i\omega(a_i^2)\ , \tag{1}$$

$$\rho=(\rho_1,...,\rho_n),\ i\in I,\ \rho_i>0,\ \sum_{i\in I}\rho_i=1, \tag{2}$$

Где (2) – нормированный вектор относительной важности параметров объектов для утверждения эксперта об отношении нечеткого предпочтения между объектами.

Таким образом, задача состоит в локализации весовых коэффициентов свертки (1)-(2). Подобная задача рассматривалась в [Волошин, 2003], в настоящей работе предлагается обобщение метода для нечетких моделей выбора.

Процедуры локализации вектора весовых коэффициентов

Пусть ЛПР считает, что объект α^1 предпочтительней объекта α^2, а μ – степень предпочтения, $\mu\in[0,1]$. Обозначим это через $\mu_{\succ}\left(a^1,a^2\right)$. Обозначим также $1-\mu_{\succ}\left(a^1,a^2\right)$ степень предпочтения объекта α^2 над объектом α^1. Введем эвристику: будем считать, что из субъективного суждения ЛПР о предпочтении объекта α^1 над α^2 следует справедливость неравенства (1). Тогда для случая нечеткого отношения предпочтения $\mu_{\succ}\left(a^1,a^2\right)$ между объектами α^1 и α^2 будем считать справедливым:

$$\mu_{\succ}\left(a^1,a^2\right) \Leftrightarrow \sum_{i\in I}\rho_i\frac{\omega(a^1{}_i)}{\mu}\le\sum_{i\in I}\rho_i\frac{\omega(a^2{}_i)}{1-\mu}\ . \tag{3}$$

Необходимо построить на основе отношений предпочтений на множестве эффективных объектов A, которые последовательно уточняются ЛПР, интервалы допустимых значений весовых коэффициентов параметров объектов (гиперпараллелепипед весовых коэффициентов - ГВК) в виде

$$\rho\in K=\prod_{i\in I}[\rho_i^H,\rho_i^B],\quad \rho=(\rho_i,\ i\in I),\quad 0<\rho_i^H\le\rho_i^B<1, \tag{4}$$

$$\sum_{i\in I}\rho_i=1,\rho_i>0,i\in I\ . \tag{5}$$

Предполагается, что при задании предпочтений ЛПР последовательно в своих суждениях, в частности, задаёт отношения предпочтения между объектами, которые удовлетворяют свойству транзитивности. Поскольку попарное сравнение объектов производится с помощью линейной свертки (1) и ГВК в результате работы предлагаемой процедуры пошагово сужается $(K^{S+1}\subseteq K^S,\ s=1,2,...)$, то транзитивность задаваемого бинарного отношения сохраняется.

Для преобразования всех значений параметров объектов $a^j{}_i$, $i\in I$, $j\in J$, к безразмерному виду в интервале [0,1] применим формулу:

$$\omega(a^j{}_i)=\frac{a^{opt}{}_i-a^j{}_i}{a^{opt}{}_i-a^0{}_i}\ , \tag{6}$$

где $\alpha^j{}_i\in A$, $i\in I$, $j\in J$; $\alpha^{opt}{}_i\in A$, $i\in I$ - наилучшее значение i-го параметра на множестве эффективных объектов: $\alpha^0{}_i\in A$, $i\in I$ - наихудшее значение i-го параметра на множестве эффективных объектов. Будем считать, что α^{opt} и α^0 могут быть заданы непосредственно ЛПР, или найдены как максимальные (минимальные) значения параметров, которые достигаются на множестве допустимых решений.

С учетом (6), обобщенный критерий, который отображает суммарное отклонение j-го объекта, $j\in J$, от оптимальных значений, запишется как

$$D(a^j, a^{opt}) = \sum_{i \in I} \rho_i \omega(a^j{}_i) = \sum_{i \in I} \rho_i \frac{a^{opt}{}_i - a^j{}_i}{a^{opt}{}_i - a^0{}_i}, \quad j \in J .$$

Последняя формула является метрикой близости вектора значений параметров объекта $\alpha^j \in A$, $j \in J$, к некоторому идеальному (оптимальному) вектору значений $\alpha^{opt} = (\alpha^{opt}_1, \alpha^{opt}_2, \ldots \alpha^{opt}_n)$, взвешенных в пространстве параметров. Формула (3) для нечеткого отношения предпочтения между объектами $\mu_{\succ}(a^1, a^2)$ запишется в виде

$$\frac{D(a^1, a^{opt})}{D(a^2, a^{opt})} = \frac{\sum_{i \in I} \rho_i \omega(a^1{}_i)}{\sum_{i \in I} \rho_i \omega(a^2{}_i)} \le \frac{\mu}{1-\mu} .$$

Последнее неравенство можно интерпретировать таким образом: утверждение "объект α^1 предпочтительней чем объект α^2 со степенью предпочтительности μ" обозначает, что в пространстве параметров объектов точка, которая соответствует объекту α^1 находится ближе к идеальной точке, чем точка, которая соответствует объекту α^2 со степенью $\frac{\mu}{1-\mu}$.

Определение 1. Объекты $a^1 \in A$ и $a^2 \in A$ называются равноценными, если во "взвешенном" пространстве параметров Ω^n, соответствующие им точки находятся на одинаковом расстоянии от точки, соответствующей идеальному объекту.

Определение 2. Объект $a^2 \in A$ называется μ-равноценным объекту $a^1 \in A$, если во "взвешенном" пространстве параметров Ω^n точка

$\omega(a^2)\frac{\mu}{1-\mu} = \left\{ \omega(a^2{}_1)\frac{\mu}{1-\mu}, \omega(a^2{}_2)\frac{\mu}{1-\mu}, \ldots, \omega(a^2{}_n)\frac{\mu}{1-\mu} \right\}$ и точка $\omega(a^1)$ равноценны.

Утверждение 1. Весовые коэффициенты параметров $\rho = (\rho_i,\ i \in I)$, соответствующие μ- равноценным объектам, в пространстве предпочтений R^n отделяют уточненные границы интервалов весовых коэффициентов параметров объектов.

Доказательство. Обозначим множества индексов параметров объектов $a^1 \in A$ и $a^2 \in A$ через $I_1 = (i : \omega(a^1{}_i) > \omega(a^2{}_i)) \neq 0$, $I_2 = (i : \omega(a^1{}_i) \le \omega(a^2{}_i)) \neq 0$, $i \in I = I_1 \cup I_2$. Неравенство (2) можно переписать с учетом уточнения множеств индексов следующим образом:

$$\sum_{\substack{i \in I_1 \\ \rho_i^s \in K^s}} \rho_i^s \omega(a_i^1) + \sum_{\substack{i \in I_2 \\ \rho_i^s \in K^s}} \rho_i^s \omega(a_i^1) \le \frac{\mu}{1-\mu} \sum_{\substack{i \in I_1 \\ \rho_i^s \in K^s}} \rho_i^s \omega(a_i^2) + \frac{\mu}{1-\mu} \sum_{\substack{i \in I_2 \\ \rho_i^s \in K^s}} \rho_i^s \omega(a_i^2) . \quad (7)$$

Тогда условие μ- равноценности объектов α^1 и α^2 запишется как

$$\sum_{\substack{i \in I_1 \\ \rho_i^s \in K^s}} \rho_i^s \omega(a_i^1) + \sum_{\substack{i \in I_2 \\ \rho_i^s \in K^s}} \rho_i^s \omega(a_i^1) = \frac{\mu}{1-\mu} \sum_{\substack{i \in I_1 \\ \rho_i^s \in K^s}} \rho_i^s \omega(a_i^2) + \frac{\mu}{1-\mu} \sum_{\substack{i \in I_2 \\ \rho_i^s \in K^s}} \rho_i^s \omega(a_i^2) . \quad (8)$$

Перейти от (7) к (8) можно, увеличив весовые коэффициенты параметров, которые принадлежат множеству индексов I_1 и, соответственно, уменьшив весовые коэффициенты параметров, которые принадлежат множеству индексов I_2. Таким образом, весовые коэффициенты параметров ρ_i, $i \in I_1$, достигнут своих верхних границ, а весовые коэффициенты параметров, ρ_i, $i \in I_2$, соответственно достигнут своих нижних границ.

Поскольку $\omega(\alpha^1_i)$, $\omega(\alpha^2_i)$, $i \in I$, являются фиксированными величинами, а $\rho^s_i \in K^s$, то полученное равенство можно записать как

$$\sum_{\substack{i\in I_1\\ \rho_i^s\in K^s}}\rho_i^{(s)B}\omega(a_i^1)+\sum_{\substack{i\in I_2\\ \rho_i^s\in K^s}}\rho_i^{(s)H}\omega(a_i^1)=\sum_{\substack{i\in I_1\\ \rho_i^s\in K^s}}\rho_i^{(s)B}\omega(a_i^2)+\sum_{\substack{i\in I_2\\ \rho_i^s\in K^s}}\rho_i^{(s)H}\omega(a_i^2)\,. \qquad (9)$$

Распространяя (9) на случай μ- равноценности объектов α^1 и α^2 , получим окончательно

$$\sum_{\substack{i\in I_1\\ \rho_i^s\in K^s}}\rho_i^{(s)B}\omega(a_i^1)+\sum_{\substack{i\in I_2\\ \rho_i^s\in K^s}}\rho_i^{(s)H}\omega(a_i^1)=\frac{\mu}{1-\mu}\sum_{\substack{i\in I_1\\ \rho_i^s\in K^s}}\rho_i^{(s)B}\omega(a_i^2)+\frac{\mu}{1-\mu}\sum_{\substack{i\in I_2\\ \rho_i^s\in K^s}}\rho_i^{(s)H}\omega(a_i^2)\,, \qquad (10)$$

где $\rho^{(s)B}_i$, $\rho^{(s)H}_i$, $i\in I$, - соответственно верхняя и нижняя границы i-го интервала весовых коэффициентов на s-м шаге алгоритма. Равенство (10) эквивалентно равенству (8).

Таким образом, ГВК на s+1 шаге станет равным

$$K^{s+1}=\prod_{i\in I_1}[\rho_i^{(s)H},\rho_i^{(s+1)B}]\times\prod_{i\in I_2}\left[\rho^{(s+1)H},\rho^{(s)B}\right], \qquad (11)$$

Уравнение (9) доказывает справедливость утверждения 1: найденные применением описанного метода весовые коэффициенты действительно определяют границы ГВК.

Поскольку известен лишь факт предпочтения ЛПР, заданный в форме (3), то для определения компонент вектора $\rho=(\rho_i,\ i\in I)$, сделаем предположение о справедливости k неравенств вида:

$$\rho_i\omega(a^1{}_i)+\rho_j\omega(a^1{}_j)\le\frac{\mu}{1-\mu}(\rho_i\omega(a^2{}_i)+\rho_j\omega(a^2{}_j))\,, \qquad (12)$$

где $k=k_1\cdot k_2$; k_1 – количество параметров с индексами i, $i\in I_1$; k_2 – количество параметров с индексами j, $j\in I_2$.

Очевидно, что выполнение системы неравенств (12) является достаточным условием для выполнения неравенства (7).

Далее перейдем в системе уравнений (12) к равенствам и исключим k-n-1 равенство по правилу: каждый раз исключается равенство, которое доставляет максимум выражению

$$\max((\omega(a^1{}_i)-\omega(a^2{}_i),\ i\in I_1,\ \omega(a^2{}_j)-\omega(a^1{}_j),\ j\in I_2)\,.$$

Последнее условие обозначает, что отбрасываются равенства, которые создают неоправданно большие приращения весов одних параметров объектов за счет других.

Добавим в систему n-1 неравенств вида (12) в качестве n-го равенства условие нормирования весовых коэффициентов (5), перейдем от них к равенствам и, проведя некоторые преобразования, окончательно получим систему n уравнений вида:

$$\rho_{\ i\in I_1}(\omega(a^1{}_i)-\frac{\mu}{1-\mu}\omega(a^2{}_i))-\rho_j(\frac{\mu}{1-\mu}\omega(a^2{}_j)-(\omega(a^1{}_i))=0\,, \qquad (13)$$

$$\sum_{i\in I}\rho_i=1,\rho_i>0,i\in I\,.$$

Из системы уравнений (13) однозначно определяются компоненты вектора весовых коэффициентов $\rho=(\rho_i,\ i\in I)$, который по утверждению 1 ограничивает в векторном пространстве R^n интервалы весовых коэффициентов параметров объектов.

С учетом описанного выше очевидна справедливость следующего утверждения.

Утверждение 2. Условием отсеивания объектов ω^j, $j\in J$, из множества A^s является непринадлежность ГВК вектора, который проходит через начало координат и точку $\omega(\alpha^j)$, $\alpha^j\in A^s$, $j\in J$, то есть $\rho(\omega(a^j))\notin K^{(s+1)}$. Вектор весовых коэффициентов определяется по формуле, приведенной в [4]:

$$\rho=\rho(\omega(a^j))=\{\rho_i:\rho_i=\prod_{\substack{t\in I\\ t\ne i}}\omega(a^j{}_t)/\sum_{\substack{q\in I\\ l\ne q}}\prod_{l\in I}\omega(a^j{}_l)\}\,.$$

Человеко-машинная процедура определения гиперпараллелепипеда весовых коэффициентов описывается в виде такой последовательности шагов.

Шаг 1. Выделение множества эффективных объектов A^0 из заданного множества объектов А одним из методов, которые приводятся в работе [Волкович, 1993]. Первоначальный ГВК полагается равным единичному гиперкубу.

Шаг 2. Выбор ЛПР двух объектов α^1 и α^2 из множества эффективных объектов A^s в ГВК K^s, s =1,2,... (шаг сужения ГВК) с указанием факта предпочтения или эквивалентности.

Шаг 3.Построение системы уравнений вида (13). Нахождение решения системы уравнений.

Шаг 4. Уточнение границ ГВК по формуле (11). Если гиперкуб $K^{(s+1)}$ удовлетворяет ЛПР, то окончание процедуры. Иначе переход к следующему шагу.

Шаг 5. Выделение множества эффективных объектов $A^{(s+1)}$ ($A^{(s+1)} \subseteq A^{(s)}$) в ГВК $K^{(s+1)}$ и предъявление их ЛПР для выбора очередных двух объектов и указания для них отношения предпочтения. Увеличение номера итерации: s=s+1. Переход к шагу 2.

Выводы

Предложенные процедуры, не требуя полной матрицы парных сравнений объектов, позволяют на множестве нечетких бинарных отношений восстановить функцию полезности эксперта. Отображение вектора весовых коэффициентов в виде интервалов позволяет адекватно представлять уровень неопределенности в нечетких моделях принятия решений.

Библиография

[Ларичев, 2002] Ларичев О.И. Свойства методов принятия решений в многокритериальных задачах индивидуального выбора // Автоматика и телемеханика, № 2, 2002. – С. 146-158.

[Борисов, 1989] Борисов А.Н., Алексеев А.В. и др. Обработка нечеткой информации в системах принятия решений. – М: Радио и связь, 1989. – 304 с.

[Пономаренко, 1994] Пономаренко О.І. Системні методи в економиці, менеджменті та бізнесі. – К.: Наукова думка, 1994. – 242 с.

[Волошин, 2003] Волошин А.Ф., Гнатиенко Г.Н., Дробот Е.В. Метод косвенного определения интервалов весовых коэффициентов параметров для метризированных отношений между объектами // Проблемы управления и информатики, 2003, № 2.

[Волкович, 1993] Волкович В.Л., Волошин А.Ф., др. Модели и методы оптимизации надежности сложных систем / Под ред. Михалевича В.Ф. – К.: Наукова думка, 1993. – 312 с.

Сведения об авторе

Елена Присяжнюк *– Кировоградский педагогический университет имени В.Винниченко, к.т.н., доцент, Кировоград, Украина, e-mail: elena_drobot@ukr.net*

РАЗРАБОТКА КАТАЛОГА МЕТАДАННЫХ СИСТЕМЫ GEO-UKRAINE

Наталья Куссуль, Алина Рудакова, Алексей Кравченко

***Аннотация**: Статья посвящена актуальному вопросу каталогизации разнородных данных – дистанционного зондирования, наземных измерений и данных моделирования. В работе предлагается концептуальный подход к созданию системы каталогизации, с учетом мирового опыта построения таких систем.*

Рассмотрены архитектурные особенности разрабатываемого каталога метаданных, существующие решения для каталогов метаданных, вопросы стандартизации метаданных и создания профилей метаданных, а также некоторые аспекты реализации каталога метаданных в информационной системе GEO-Ukraine.

***Ключевые слова**: каталог данных, метаданные, профиль метаданных, стандарт ISO 19115*

Введение

Современные спутниковые данные представляют особый интерес для решения разнообразных задач из областей науки, сельского хозяйства и др., в целях снижения рисков стихийных бедствий, оценки последствий природных и техногенных катастроф. Данные дистанционного зондирования (ДЗЗ) по сравнению с данными наземных наблюдений обладают рядом преимуществ, в т.ч. стоимость, доступность, оперативность. Поэтому круг потребителей аэрокосмической информации в Украине и в мире расширяется.

В Украине достаточно большое количество организаций занимается решением разнообразных тематических задач, связанных с анализом данных ДЗЗ. По мере решения подобных задач в каждой организации накапливается собственный архив геопространственных данных, эффективное использование которого невозможно без его каталогизации.

В Украине разрабатывается информационная система GEO-Ukraine [1], которая должна стать украинским сегментом в международной системе систем наблюдений Земли GEOSS (www.earthobservations.org). В рамках системы GEO-Ukraine (http://www.geoss-ukraine.org.ua) одной из приоритетных задач является объединение уже существующих баз геопространственных данных. Однако они имеют сложную структуру, разные форматы, часто большой объем, не позволяющий активно использовать данные в сети Internet. Залогом эффективного использования данных ДЗЗ является инфраструктура хранения, поиска и предоставления этих данных на основе их структурированных описаний – метаданных.

Метаданные предоставляют необходимый и достаточный объем информации необходимый для того, чтобы понять природу и суть содержания описываемого набора данных. В данное время метаданные являются неотъемлемой частью при создании хранилищ данных и их каталогизации. Создание такого каталога метаданных упрощает управление, создание запросов, полноценное использование и понимание данных. Для унификации процессов поиска и предоставления информации в рамках единой системы необходима стандартизация наборов и форматов предоставления метаданных, т.е. необходимо разработать профиль метаданных.

Существующие решения

Проблема создания каталогов метаданных стала актуальной в связи с взрывообразным увеличением объемов данных, поступающих со спутников ДЗЗ. На данный момент каталоги метаданных созданы ведущими национальными космическими агентствами, поставщиками и дистрибьюторами данных ДЗЗ, научными организациями, работающими со спутниковыми данными. Среди существующих зарубежных каталогов метаданных следует упомянуть о каталоге данных ДЗЗ Европейского космического агентства (ESA) (http://earth.esa.int/resources/catalogues/) и каталоге NASA EOS Data Gateway (http://delenn.gsfc.nasa.gov/~imswww/pub/imswelcome/).

Каталог метаданных ESA Multi-mission Catalogue Service позволяет пользователям выполнять поиск данных ДЗЗ, отображать результаты поиска на карте, анализировать изображения для предварительного просмотра. При поиске имеется возможность указывать необходимый набор спутников и сенсоров, а также задавать временные и пространственные характеристики снимков. Поддерживается два вида пользователей – анонимные и авторизированные. Авторизированным пользователям предоставляется дополнительная услуга заказывать данные ДЗЗ в режиме online. Каталог создан для предоставления данных как со спутников ESA (ERS-1/2, Envisat), так и других миссий (Landsat, PROBA, SPOT). Для этого каталога создано несколько интерфейсов пользователя, в частности Web-интерфейс EOLI-Web (http://eoli.esa.int/servlets/template/welcome/entryPage2.vm), и настольное приложение EOLI-SA. Оба приложения созданы на основе технологии Java.

Каталог EOS Data Gateway является распределенной системой поиска и заказа данных ДЗЗ и результатов их обработки. Интерфейс пользователя каталога реализован в виде тонкого Web-клиента. Подсистема поиска позволяет указывать наборы данных, временные и пространственные ограничения для продуктов ДЗЗ, а также некоторые дополнительные параметры. Пользователь может просматривать атрибуты снимков, предварительные изображения данных, а также метаинформацию о наборах данных, спутниках и сенсорах. Каталог предоставляет доступ к данным NASA, в основном к данным приборов на борту спутников Aqua и Terra. Система поддерживает анонимных (guests) и зарегистрированных пользователей. Последние имеют возможность сохранять параметры поисковых запросов и результатов поиска, а также повторно использовать регистрационную информацию при заказе данных.

Среди украинских каталогов метаданных следует упомянуть каталог метаданных, созданный в ГНПЦ «Природа» и в Центре аэрокосмических исследований Земли (ЦАКИЗ). Каталог ГНПЦ «Природы» доступен в Internet по адресу http://www.pryroda.gov.ua/pryroda/search_film.do и позволяет осуществлять поиск спутниковых данных, которыми располагает организация. Каталог создан для облегчения поиска данных клиентами для заказа продукции этой организации. Поисковая система каталога позволяет осуществлять поиск данных, указывая конкретный спутник или сенсор, временные и пространственные ограничения для снимка. Имеется возможность просматривать местоположения снимков на карте и изображения снимков для быстрого просмотра. Каталог ЦАКИЗ на данный момент доступен только в Intranet-сети этой организации и позволяет проводить поиск по атрибутам снимка (спутник, сенсор), временным и пространственным параметрам снимка. Отдельно существует интерфейс внесения новых метаданных. Каталог создан для поддержки работ внутри самой организации. В Украине достаточно большое число организаций занимается тематической обработкой данных ДЗЗ и имеет накопленный архив данных, который зачастую не структурирован. Для эффективного использования существующих данных ДЗЗ необходимо создание каталога метаданных о данных ДЗЗ. Модели метаданных данных каталогов будут рассмотрены далее.

Организация инфраструктуры пространственных данных и их хранения определяются стандартами. Они задают язык и правила взаимодействия участников, без которых это взаимодействие невозможно.

В данное время разработано множество стандартов по представлению пространственных данных. Ряд международных организаций занимается проблемой стандартизации метаданных геопространственных данных.

Среди ряда международных организаций занимающихся стандартизацией геопространственных данных и геоинформационных систем необходимо отметить Федеральный комитет по географическим данным FGDC (http://www.fgdc.gov/) разработавший один из первых стандартов для метаданных Content Standard for Digital Geospatial Metadata. Этот стандарт использовался при построении Национальной инфраструктуры геопространственных данных службой США.

Организация Open GIS Consortium (OGC) была основана в 1994 году с целью «обеспечения спецификаций пространственного интерфейса, доступных для всеобщего использования». Основная цель консорциума OGC — создание технологий обеспечивающих прозрачность взаимодействия программных средств разных производителей, возможность конвертирования данных разных форматов и их совместного использования, открытость функциональных интерфейсов и унификация для конечных пользователей (www.opengis.org).

Наиболее общепризнанным и используемым в большинстве международных и национальных проектах является стандарт ISO 19115 Geographic information – Metadata Международной организации по стандартизации ISO/TC 211 (www.isotc211.org). Данный стандарт рекомендован к использованию в рамках международной системы GEOSS и инициативы INSPIRE (Infrastructure for Spatial Information in Europe — Инфраструктура для пространственной информации в Европе), которая является основой для обмена геопространственными данными в рамках европейской программы GMES (www.gmes.info).

Стандарт ISO 19115

Основной задачей стандарта является определение методологии формирования метаданных для географической информации. В стандарте определена терминология, методология и универсальный набор элементов метаданных, посредством которых эти метаданные описываются.

Данный стандарт определяет:

– обязательные (О) и условные (У) пакеты метаданных, их сущности и элементы;

– необходимый и достаточный набор метаданных, для большинства случаев их использования (поиск данных, определение соответствия данных, доступ к данным, передача данных и использование цифровых данных);

– необязательные (Н) элементы метаданных, позволяющие расширить стандартное описание геопространственных данных при необходимости.

Метаданные представляются совокупностью UML-пакетов. Каждый пакет имеет смысловое наполнение и характеризует тот или иной аспект метаданных. Достоинством этого стандарта является то, что он представлен на Универсальном языке моделирования (UML), так как UML-диаграммы могут использоваться для генерации схемы базы данных в полном соответствии с этим стандартом. В стандарте определено 14 UML-пакетов, каждый из которых состоит из одной или нескольких сущностей (рис. 1). Сущность (UML-класс) формально описывает группы объектов, которые обладают одинаковым набором характеристик. Классы содержат элементы (атрибуты), характеризующие конкретный экземпляр метаданных.

Стандартом определен конкретный набор элементов метаданных, однако далеко не все из них используются при описании той или иной географической информации. Поэтому актуальной задачей является создание профиля метаданных для решения конкретной задачи.

Профиль метаданных является ключевым понятием геопространственных стандартов ISO и определяет специфический набор атрибутов метаданных, которые должны быть заполнены в описании для того, чтобы данное описание удовлетворяло данному профилю. Предусматривается, что профили метаданных могут создаваться как отдельными организациями, так и комитетами, работающими в направлении гармонизации. В стандарте полностью прописана процедура создания профиля метаданных.

В качестве основы для формирования профиля определен базовый набор элементов метаданных называемый «ядром метаданных», необходимый для основного документирования географических данных. Элементы «ядра метаданных» предоставляют минимальный объем информации, необходимый для понимания природы и содержания описываемого набора данных:

– информация о метаданных;

– информация о данных;

– информация о способе получения данных;

– информация о системе координат;

– информация о происхождении данных.

Идентификатор файла метаданных (Н) (MD_Metadata.fileIdentifier)	**Стандарт кодировки данных (О)** (MD_Metadata > MD_DataIdentification.characterSet)
Название стандарта метаданных (Н) (MD_Metadata.metadataStandardName)	**Основная тема набора данных (О)** (MD_Metadata > MD_DataIdentification.topicCategory)
Версия стандарта метаданных (Н) (MD_Metadata.metadataStandardVersion)	**Краткое содержание набора данных (О)** (MD_Metadata > MD_DataIdentification.abstract)
Язык создания метаданных (У) (MD_Metadata.language)	**Масштаб или расстояние на местности (Н)** (MD_Metadata > MD_DataIdentification.spatialResolution > MD_Resolution.equivalentScale or MD_Resolution.distance)
Стандарт кодировки метаданных (У) (MD_Metadata.characterSet)	**Метод пространственного представления (Н)** (MD_Metadata > MD_DataIdentification.spatialRepresentationType)
Сторона, ответственная за метаданные (О) (MD_Metadata.contact > CI_ResponsibleParty)	**Географическое положение набора данных, координаты или географический идентификатор (У)** (MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_GeographicExtent > EX_GeographicBoundingBox or EX_GeographicDescription)
Дата создания метаданных (О) (MD_Metadata.dateStamp)	**Пространственно-временные характеристики (Н)** (MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_TemporalExtent or EX_VerticalExtent)
Название набора данных (О) (MD_Metadata > MD_DataIdentification.citation > CI_Citation.title)	**Информация об Интернет-ресурсах (Н)** (MD_Metadata > MD_Distribution > MD_DigitalTransferOption.onLine > CI_OnlineResource)
Дата создания набора данных (О) (MD_Metadata > MD_DataIdentification.citation > CI_Citation.date)	**Формат данных и версия формата (Н)** (MD_Metadata > MD_Distribution > MD_Format.name and MD_Format.version)
Сторона, ответственная за набор данных (Н) (MD_Metadata > MD_DataIdentification.pointOfContact > CI_ResponsibleParty)	**Система координат (Н)** (MD_Metadata > MD_ReferenceSystem)
Язык создания данных (О) (MD_Metadata > MD_DataIdentification.language)	**Информация о происхождении данных (Н)** (MD_Metadata > DQ_DataQuality.lineage > LI_Lineage)

Дополняя базовый набор метаданных другими элементами, обеспечивается необходимая степень детализации в зависимости от решаемой задачи.

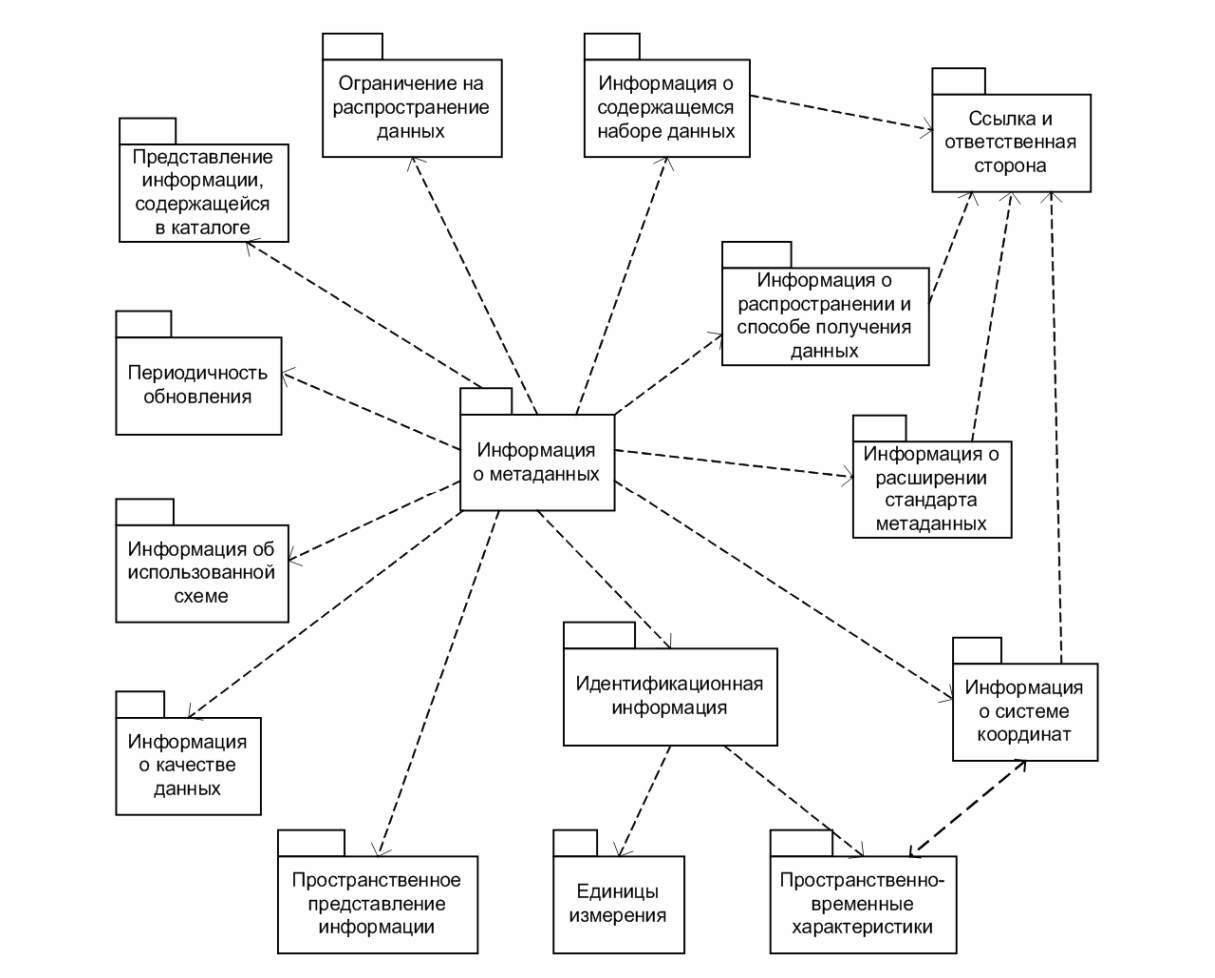


Рис. 1. UML-пакеты метаданных

Создание профиля стандарта метаданных

Профиль метаданных разрабатывается с учетом специфики предметной области в зависимости от решаемых задач и представляет собой некую «проекцию» стандарта на предметную область (рис. 2). Профиль метаданных для украинского сегмента должен быть близким к международным разработкам в этой области и соответствовать интересам Украины в лице организаций, которые являются пользователями геопространственных данных.

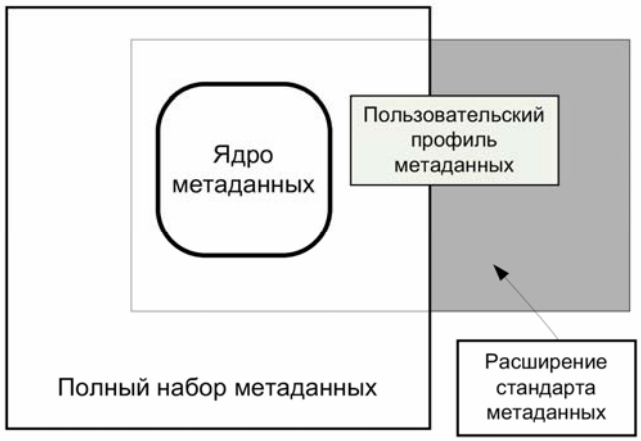


Рис. 2. Механизм создания пользовательского профиля метаданных

В стандарте ISO №19115 прописаны следующие правила создания профиля:

1. Перед созданием профиля необходимо проверить зарегистрированные профили.

2. Профиль создается в соответствии с правилами для определения расширений.

3. В профиле не изменяются имена, определения или типы данных элементов метаданных.

4. Профиль должен содержать:

– элементы ядра метаданных;

– все обязательные элементы метаданных во всех обязательных секциях;

– все условные элементы метаданных во всех обязательных секциях, если набор данных удовлетворяет условию, необходимому для включения элементов метаданных в соответствующую секцию;

– все обязательные элементы метаданных во всех условных секциях, если набор данных удовлетворяет заданному условию;

– все условные элементы метаданных во всех условных секциях, если набора данных удовлетворяет условию, необходимому для наличия элементов метаданных и секции.

5. Взаимосвязи между сущностями и пакетами метаданных должны быть определены строго в соответствии со стандартом.

6. Профиль должен быть доступным каждому пользователю, получающему и использующему метаданные созданные в соответствии с этим профилем.

Создаваемый профиль метаданных для информационной системы GEO-Ukraine должен учитывать существующие профили метаданных ГНПЦ «Природа» и ЦАКИЗ. Поля, описывающие данные этих центров присутствуют в разрабатываемом профиле в соответствии со стандартом.

В каталоге метаданных ГНПЦ «Природа» содержатся следующие атрибуты снимков:

спутник, датчик, метод регистрации изображения, процент облачности (для оптических снимков), регион снимка (в текстовом виде), дата и время съемки, информация о спектральных каналах изображения, пространственном разрешении, формате и размере данных, координаты углов снимка, и некоторые другие параметры. Дополнительно содержится RGB-изображение снимка для предварительного просмотра.

Каталог метаданных ЦАКИЗ содержит 4 основных информационных группы разработанного профиля: информация о метаданных, идентификационная информация, способ распространения данных, ссылка на владельца данных, информация об инструменте. Информация о метаданных в данном каталоге организована по иерархическому принципу, также созданы словари, которые включают весь перечень допустимых значений соответствующих атрибутов.

Необходимым атрибутом метаданных является информация об уровне обработки космических снимков. Комитетом CEOS определены уровни обработки данных L0, L1A, L1B, L2, L3, L4. В рамках этой классификации уровень обработки снимка варьируется от необработанных данных телеметрии со спутника (уровень L0) до результатов моделирования с использованием данных ДЗЗ (уровень L4).

Обязательным элементом профиля метаданных должна быть информация об изображении предварительного просмотра. При этом желательно, чтобы эти изображения содержали данные географической привязки, что позволит создать полнофункциональный интерфейс пользователя каталога метаданных.

Каталог метаданных

Каталог метаданных, разрабатываемый в Институте космических исследований, представляет собой единое хранилище метаданных о данных ДЗЗ системы GEO-Ukraine. В будущем каталог также будет предоставлять информацию о данных измерений in-situ, результатах тематической обработки данных ДЗЗ и результатах моделирования. Создание такого каталога будет способствовать объединению усилий в

рамках системы GEO-Ukraine, создаваемой под эгидой Национального космического агентства Украины как национальный сегмент международной системы GEOSS [2]. В частности каталог позволит избежать дублирования данных в системе GEO-Ukraine, существенно упростит обработку данных ДЗЗ, распределенную между несколькими организациями (в первую очередь автоматическую), упростит заказ услуг на обработку данных ДЗЗ.

В результате анализа потребностей сообщества ДЗЗ в Украине было выделено следующие группы пользователей каталога метаданных:

- представители сообщества ДЗЗ в Украине, желающие получить доступ к данным ДЗЗ;
- представители поставщиков данных ДЗЗ и продуктов их обработки;
- автоматизированные и полностью автоматические системы обработки данных.

Исходя из выделенных групп пользователей, определены основные прецеденты использования каталога:

- интерактивный поиск данных;
- автоматический поиск данных;
- внесение метаданных в систему.

Сообщество GEO-Ukraine состоит из территориально распределенных организаций, которые занимаются обработкой данных ДЗЗ, центров приема данных ДЗЗ, обработкой и хранением. В такой распределенной системе существуют два подхода организации метаданных в единое хранилище: централизованный и децентрализованный.

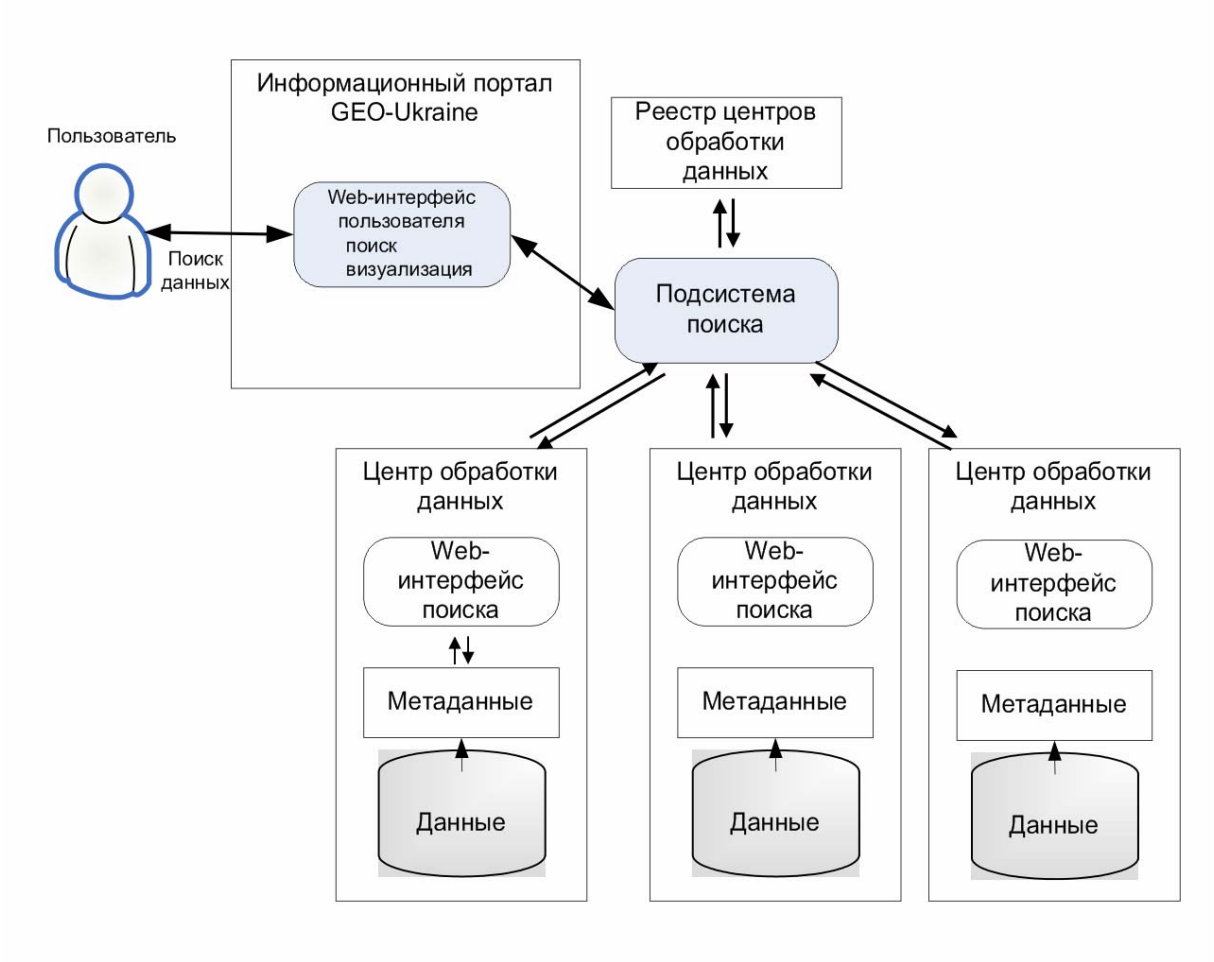


Рис. 3. Децентрализованная схема организации метаданных

Реализация децентрализованного способа и соответствующей архитектуры единого хранилища метаданных показаны на рис. 3. В рамках децентрализованного подхода метаданные хранятся непосредственно в организациях, которые хранят эти данные. В рамках каталога метаданных ведется реестр таких центров и создана распределенная подсистема поиска данных. Для организации подсистемы поиска данных определена и стандартизирована модель метаданных, в каждом центре обработки данных создана локальная подсистема поиска данных, а также специфицирован интерфейс к поисковым системам для каждого центра. В центральной подсистеме поиска созданы адаптеры подсистем поиска для каждого центра обработки данных. Создание такой системы существенно облегчает стандартизация поисковых подсистем центров обработки данных. Для взаимодействия внешних пользователей с каталогом метаданных может быть создан единый интерфейс, в частности на Web-технологиях. По такой схеме устроен каталог метаданных EOS Data Gateway.

Достоинством данной схемы является высокая скорость обновления метаданных, поскольку они вносятся локально каждым центром обработки данных. Однако она обладает существенными недостатками, а именно:

- скорость поиска лимитирована наихудшими показателями параметров связи между центральной подсистемой поиска и центрами обработки данных, а также наихудшей производительностью подсистем поиска метаданных;
- организациям, которые желают быть представленными в каталоге, необходимо создать и поддерживать информационную инфраструктуру, в частности обеспечивать хранение метаданных, поиск по метаданным, обеспечивать надежность предоставления услуг.

В рамках централизованной схемы метаданные обо всех данных в системе хранятся в выделенном хранилище. Данная схема поддерживает несколько типов центров обработки данных (рис. 4):

- центры, поддерживающих собственное хранилище метаданных;
- центры первого типа, которые дополнительно обеспечивают поисковый интерфейс пользователя к собственному хранилищу метаданных;
- центры, не имеющие собственного хранилища метаданных.

Для работы с центрами первого и второго типов должен быть определен протокол сбора метаданных для каждого центра. Для поддержки центров третьего типа должен быть создан специализированный интерфейс, предназначенный для внесения пользовательских метаданных в каталог.

Данная схема не обладает недостатками предыдущей благодаря возможности обновления метаданных в режиме offline и поддержки нескольких типов центров обработки данных. В то же время частота обновления сведений о данных ограничена частотой сбора информации с центров обработки данных.

По данной схеме организованы большинство поисковых систем общего назначения, в частности поисковые системы Google (www.google.com) и Yahoo (www.yahoo.com).

При создании каталога метаданных для системы GEO-Ukraine необходимо учитывать следующие особенности украинских организаций, занимающихся обработкой данных ДЗЗ:

- различное состояние информационной инфраструктуры в подобных организациях;
- существование организаций, для которых создание соответствующей инфраструктуры нецелесообразно (например, в случае небольших объемов данных);
- недостаточная обеспеченность многих организаций необходимым доступом в Internet.

Учитывая эти особенности и преимущества централизованного похода к организации метаданных, для реализации каталога метаданных системы GEO-Ukraine была выбрана централизованная схема каталога метаданных.

В докладе будут рассмотрены рекомендации для представления и хранения пространственных метаданных, разработанные в Институте космических исследований НАНУ-НКАУ на основе стандарта ISO 19115.

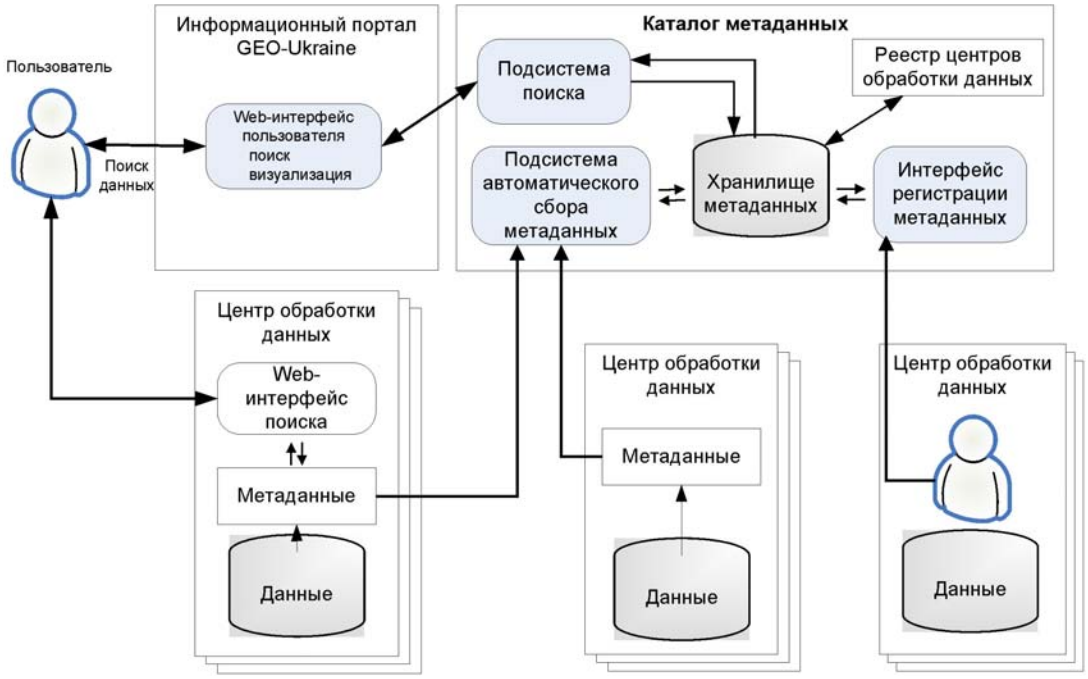


Рис. 4. Централизованная схема организации метаданных

Выводы

В данной статье предлагается концептуальный подход к созданию системы каталогизации, с учетом мирового опыта построения таких систем.

Каталоги данных и метаданных системы GEO-Ukraine планируется внедрить в международный каталог метаданных ECHO (Earth Observing System Clearing House, www.echo.eos.nasa.gov).

Работа выполнена при поддержке проектов УНТЦ №3872 «Разработка эффективных GRID-технологий экологического мониторинга на основе спутниковых данных» и INTAS-CNES-NSAU №06-1000024-9154 «Data Fusion Grid Infrastructure».

Литература

[1] Федоров О.П., Куссуль Н.Н., Шелестов А.Ю. Задачи и перспективы развития в Украине информационной системы наблюдения Земли из космоса// Проблемы управления и информатики. — 2006. — №6. — С. 116–121.

[2] Global Earth Observation System of Systems GEOSS. 10-Year Implementation Plan Reference Document. Noordwijk, Netherlands: ESA Publication Division. 2005. 212 p

Авторы

Куссуль Наталья Николаевна *- Институт космических исследований НАН и НКА Украины, профессор, доктор технических наук, заведующий отделом, проспект Академика Глушкова 40, 03680 Киев, Украина; e-mail: inform@ikd.kiev.ua*

Рудакова Алина Игоревна *- Институт космических исследований НАН и НКА Украины, младший научный сотрудник; проспект Академика Глушкова 40, 03680 Киев, Украина; e-mail: alinarudakova@yahoo.com*

Кравченко Алексей Николаевич *- Институт космических исследований НАН и НКА Украины, аспирант, младший научный сотрудник; проспект Академика Глушкова 40, 03680 Киев, Украина; e-mail: oleksiy.kravchenko@gmail.com*

AGENT-BASED ANOMALIES MONITORING IN DISTRIBUTED SYSTEMS

Andrii Shelestov

Abstract. *In this paper an agent-based approach for anomalies monitoring in distributed systems such as computer networks, or Grid systems is proposed. This approach envisages on-line and off-line monitoring in order to analyze users' activity. On-line monitoring is carried in real time, and is used to predict user actions. Off-line monitoring is done after the user has ended his work, and is based on the analysis of statistical information obtained during user's work. In both cases neural networks are used in order to predict user actions and to distinguish normal and anomalous user behavior.*

Keywords: *security, distributed systems, agent approach, neural networks.*

ACM Classification Keywords*: K.6.5 Security and Protection – Authentication, I.2.6 Learning - Connectionism and neural nets, I.2.11 Distributed Artificial Intelligence - Multiagent systems.*

1 Introduction

Nowadays it is practically impossible to imagine different areas of human activity without the use of distributed systems, for example, corporate computer networks, Grid systems [1] for complex scientific problems solving, etc. However, it is evident that the work of many organizations (or set of organizations) considerably depends upon effective use of distributed systems resources and the level of their protection. Many problems, such as data storage, data transfer, information processing automation, complex problems solving are entrusted on them. The security level of information used in distributed systems can vary from private and business to military and state secret. The violation of information confidentiality, integrity and accessibility may have significant and undesirable consequences to its owner. Besides, many sources report that the majority (80%) of information security incidents is perpetrated by insiders (Microsoft Encyclopedia of Security, 2003) [2]. This means that internal computer users constitute the largest threat to the computer systems security.

Unfortunately, traditional methods (such as identification and authentication, access restriction, etc.) are not seemed to solve this problem at all. These rigorous and deterministic approaches possess some drawbacks; among them are low ability of internal malicious users detection, inability to process large amounts of information, low productivity, etc. That is why new approaches for users activity monitoring (including those relying on intelligent methods) are applied.

We may consider so called Personal Security Programs that are used by commercial companies to monitor the activity of their employees. The results of such monitoring can be used to reveal malicious users in the case of information leakage, or to find out whether users use computers for their personal purposes. For example, such programs as PC Spy (www.softdd.com/pcspy/index.htm), Inlook Express (www.jungle-monkey.com), Paparazzi (www.industar.net) allow to capture and save screen images (screenshots) showing exactly what was being viewed by users. All screens can be captured, including Web pages, chat windows, email windows, and anything else shown on the monitor. However, these programs have some disadvantages; among them are high volume of stored information and manual configuration of snapshots frequency.

Another example refers to Intrusion Detection Systems (IDS), particularly anomaly detection in computer systems. Usually, a model of normal user behavior is firstly created, so during monitoring any abnormal activity can be regarded as potential intrusion, or anomaly. Different approaches are applied to the development of anomaly detection systems: statistical methods [3], expert systems [4], finite automata [5], neural networks [6-8], agent-based systems [9], etc.

Generally, the development of monitoring system involves two phases: creation of user behavior model (normal or usual) and system implementation. First phase involves the following steps: data collection and data pre-

processing, when useful information about user activity is collected from log-files; data processing, when feature extraction is made to data representation and dimension reduction methods are used to reduce the size of the data; application of different techniques to obtain interesting characteristics of users' behavior; interpretation of the results. During the implementation phase it should be taken into account the distributed and heterogeneous nature of distributed systems and a great number of users in it. Therefore, it is advisable to provide an autonomous module for each user behavior model developed within the first phase. Moreover, in some cases this module has to move in the system since the user can work on different workstations (computers). Thus, the monitoring system has to be distributed and scalable, it should enable the work with different operating systems and data formats, it should have independent modules to enable autonomy and mobility. To meet these requirements, agent technology represents the most appropriate way [10-11].

In this paper we present an agent-based approach for anomalies monitoring in distributed systems. This approach envisages on-line and off-line monitoring that enables the detection of anomalies and irregularities in users' behavior. On-line monitoring is carried in real time, and is used to predict user actions. For this purpose, we use feed-forward neural networks [12]. Off-line monitoring is done after the user has ended his work, and is based on the analysis of statistical information obtained during user's work. We use neural network as classifier to distinguish normal and anomalous user behavior. The use of on-line and off-line monitoring allows one to reflect both dynamical and statistical features of user's activity. Considering system implementation, we use Java programming language and Aglets Software Development Kit (ASDK) for the development of mobile agents.

2 Agent Paradigm

The main point about agents is that they are autonomous, i.e. capable of acting independently. An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors [10]. That is, the agent can be characterized by the following set:

$$<S, Prog, Eff, Arch, P, A, G, E> \quad (1)$$

where E defines the environment where agent works; S — sensors through which it perceives information from environment; Eff — effectors through which agent can act on environment; P — what kind of information agent can perceive from its sensors; A — what kind of actions agent can make using its effectors; Prog (program) Prog: P–>A — defines agent's response to its percepts; G — goal the agent trying to reach; Arch — agent's architecture.

Main agents' properties are the following ones [13]: autonomy, reactivity (provides an ongoing interaction with its environment, and responds to changes that occur in it), proactiveness (means goal directed behavior of agent), social ability (ability to interact with other agents via some kind of agent-communication language, and perhaps co-operate with others), mobility (the ability of an agent to move around an electronic network), rationality (agent will act in order to achieve its goals), learning/adaptation (agents improve performance over time).

In this paper, software agents will be used for implementation of intelligent security system. In general, they represent computer programs and act in computer systems. Thus, according to (1) for software agent we have — E=computer system, Arch=program code, S and Eff represent some functions (or, in general case, programs) through which agent can interact with environment.

3 System Architecture and Functionality

The proposed intelligent security system for users' activity monitoring in distributed systems consists of the following components (Fig. 1):

— On-line User Agent that provides on-line monitoring,

— Off-line User Agent that provides off-line monitoring,

— Controller Agent that manages other agents,

— Database.

On-line User Agent. This agent is functioning in real time with aim to detect anomalies and irregularities in computer users' activity. It predicts user actions on the basis of previous ones. For this purpose a neural network is used. The output of the neural network is compared to real actions performed by user. If the relative number of correctly predicted actions larger than specified threshold, then it can be assumed that the user behavior is normal. Otherwise, it is abnormal. Additionally, this agent collects information about user's activity and stores it in database. This type of agents should be constructed for different operating systems used in computer system (e.g. Win2K/XP, Win98, FreeBSD).

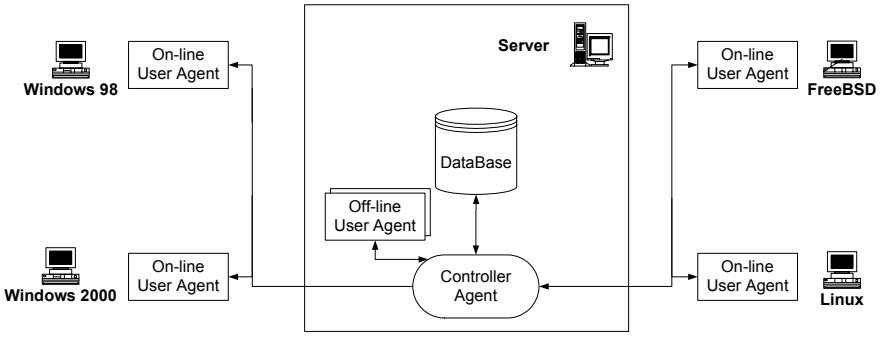


Fig. 1. System architecture

Off-line User Agent. This agent works off-line (i.e. after the user has ended his work) and tries to detect anomalies in the user activity on the basis of statistical parameters (user signature). The following set of characteristics about user behavior were taken as user signature: the set of processes (number of processes started by user), results of on-line agent functioning (number of correctly predicted processes by On-line User Agent), user login host (the set of hosts from which user logs on), user session time (the session duration for the user), user activity time (the time of user session starting). For each user its own Off-line User Agent is created based on feed-forward neural network. The network is trained in order to distinguish normal and abnormal user behavior.

Controller Agent. This agent is responsible for overall system functioning, agents initializing and coordination, and interaction with database.

Database. Contains data that is needed for system functioning.

When the user logs on (that is, begins his work on computer), Controller Agent creates corresponding On-line User Agent and initializes it. On-line User Agent gets data about specified user from a database and moves to the computer where the user works. During the user's session, this agent monitors user's activity by predicting his actions (using neural network) and comparing them to real ones. If the relative number of correctly predicted actions larger than specified threshold, then it can be assumed that user behavior is normal and corresponds to the previously built model. Otherwise, user behavior is assumed to be abnormal. In the case of anomaly detection On-line User Agent informs Controller Agent about suspicious activity. When user finishes his work, On-line User Agent is destroyed.

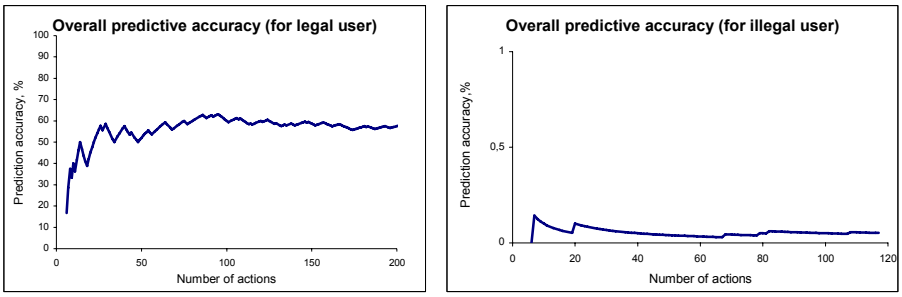
At the end of the day (when the system load is low) Controller Agent initializes Off-line User Agent. On the basis of data obtained from On-line User Agent it detects if the user activity was normal or abnormal. In the case of abnormal activity (i.e. it had anomalies) Off-line User Agent informs Controller Agent about it.

4 Description of Experiments

Different experiments were run to demonstrate the efficiency of both On-line User Agent and Off-line User Agent. Since both types of agents are based on the use of neural networks data needed for neural network training were obtained during real work of users in the Space Research Institute NASU-NSAU. For this purpose special software was developed to get data about users' activity.

For On-line User Agent log files were transformed into format suitable for neural network. That is, for each user an alphabet of actions (processes) was created, and each action was assigned an identifier (decimal number). For neural network input a binary coding was applied (7 bits for each command). Feed-forward neural network trained by means of error back-propagation algorithm [12] was used in order to predict user action on the basis of 5 previous ones. Thus, the dimension of input data space for neural network was 35. In turn, for output data decimal coding was applied, and the dimension of output data space was 1. As to neural network architecture, we used neural network with 3 layers: input layer with 35 neurons, hidden layer with 35 neurons, and output layer with 1 neuron.

Then all data were randomly mixed and divided into training and test sets (70% for training and 30% for testing). Results of neural network work on test data showed that overall predictive accuracy (that is, the number of correctly predicted commands divided by total number) for different users varied from 33% to 59% (an example of overall predictive accuracy variations within number of user actions is depicted on Fig. 2,a). But the main point in constructing On-line User Agents is to ensure that they differ for different users. That is, the efficiency of On-line User Agent should be viewed not in the term of absolute value of the predictive accuracy for the user, but relative to other users. In order to demonstrate that the neural network was able to distinguish one user from another we run so called cross experiments. Two types of cross experiments were implemented. First one consisted in the following: the data obtained during the work of one user (name him illegal user) were put to neural network that was trained for another (legal user). In such a case, overall predictive accuracy of neural network hardly exceeded 5% (on Fig. 2,b it is shown an example where overall predictive accuracy was 0,05%). That is, the overall predictive accuracy decreased, at least, six times for illegal user. Such experiment modeled the situation when illegal user logged on and begun to work under the account of another user.



(a) (b)

Fig. 2. Overall predictive accuracy for: (a) legal user; (b) illegal user

The second type of cross experiments was carried out by inserting the data of illegal user into the data of legal one. This experiment modeled the situation when an intruder begun to work under the account of another user already logged on. In a such case, the overall predictive accuracy begun to decrease constantly, as shown on Fig.3,a. Another measure that can be used to distinguish normal and anomalous user activity is a short-time predictive accuracy. To estimate the short-time predictive accuracy we took into considerations only last actions performed by the user but not all (for example, twenty last actions). Variations of short-time predictive accuracy for both legal and illegal user are shown on Fig. 3,b. From figure it is evident that the short-time predictive accuracy for illegal user begun to decrease.

Therefore, experimental results showed the ability of neural networks to distinguish confidently normal and abnormal (anomalous) user behavior.

As with On-line User Agent, all data needed for Off-line User Agent were obtained from the log files. Then the data were encoded, divided into training and test sets, and input to neural network. Results of neural network work on test data gave 80% accuracy of correct user behavior classification. That is, experiments showed that Off-line User Agent was able to distinguish normal and abnormal (anomalous) user behavior. Additionally, Off-line User Agent can be used to verify the work of On-line User Agent.

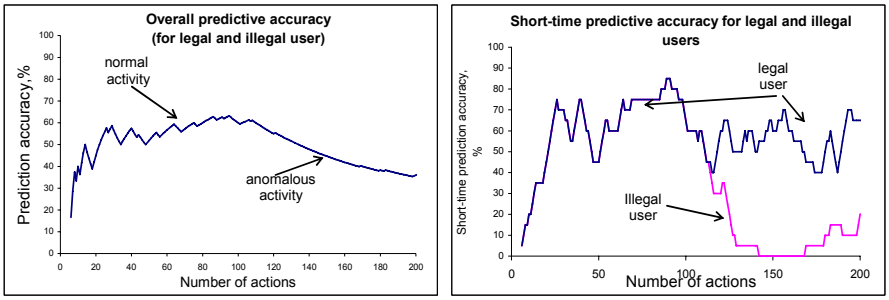


Fig. 3. Predictive accuracy for: (a) overall; (b) short-time

5 System Implementation

The proposed agent-based system was implemented using mobile agents. Java language and Aglets Software Development Kit (ASDK) were chosen, respectively, as programming language and environment for mobile agents development. Java offers the set of unique features that allows one to simplify the development of multi-agent systems. The following properties of Java should be mentioned: platform independence; secure code execution; dynamic class loading; multithreading programming; object serialization.

ASDK is a free-ware software, provided by IBM. It enables the development of mobile agents that are called aglets (http://sourceforge.net/projects/aglets/). The following properties of ASDK could be mentioned: the use of special MASIF (Mobile Agent System Interoperability Facility) standard which allows various agent systems to interoperate; the use of ATP (Agent Transfer Protocol) protocol that represents a simple application-level protocol designed to transmit an agent in an agent-system-independent manner; mobility of agents; the use of Java security policy (JDK keytool).

In general, aglets are Java objects that can move from one host on the network to another. That is, an aglet that is run on one host can suddenly halt execution, dispatch to a remote host, and start executing again. When the aglet moves, it takes along its program code as well as the states of all the objects it is carrying. A built-in special security mechanism makes it safe to host untrusted aglets.

Proposed intelligent security system was implemented based on client/server architecture. Server side represented a special platform which was used for the creation of agents and its hosting (all agents used in the system are initiated on server side), for database interaction, requests redirection. Client side is responsible for agent functioning on user computers. Among its functions are support of agents hosting and information logging about user activity. Additionally, special user interface was developed that shows information about user logged on, operating system that is used, client platform parameters, and information about On-line User Agent work.

6 Conclusions

The proposed system takes advantages of both intelligent methods for monitoring of user activity and multi-agent approach. To reflect both dynamical and statistical parameters of user behavior on-line and off-line monitoring is done. The use of neural network provides adaptive and robust approach for the analysis and generalization of data obtained during user activity. The use of multi-agent approach is motivated by the system functioning in heterogeneous environment, and by processing data in different operating systems.

Acknowledgments

The work is partly supported by the STCU and NASU Targeted Initiatives Program, project “GRID technologies for environmental monitoring using satellite data” (No. 3872) and NASU grant for Young Scientists “Development of Desktop Grid system and optimization of its productivity”.

Bibliography

[1] Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Int. J. Supercomputer Applications 15(3) (2001).

[2] Tulloch, M.: Microsoft Encyclopedia of Security. Redmond, Washington: Microsoft Press (2003) 414 p.

[3] Javitz, H., Valdes, A.: The SRI IDES statistical anomaly detector. In: Proc. IEEE Symp. on Research in Security and Privacy (1991) 316–326.

[4] Dowell, C., Ramstedt, P.: The ComputerWatch data reduction tool. In: Proc. 13th National Computer Security Conf. (1990) 99–108.

[5] Kussul, N., Sokolov, A.: Adaptive Anomaly Detection of Computer System User's Behavior Applying Markovian Chains with Variable Memory Length. Part I. Adaptive Model of Markovian Chains with Variable Memory Length. J. of Automation and Information Sciences Vol. 35 Issue 6 (2003).

[6] Ryan, J., Lin M-J., Miikkulainen, R.: Intrusion Detection with Neural Networks. In: Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press (1998) 943–949.

[7] Reznik, A, Kussul, N., Sokolov, A.: Identification of user activity using neural networks. Cybernetics and computer techniques, vol. 123 (1999) 70–79. (in Russian)

[8] Kussul, N., et al. : Multi-Agent Security System based on Neural Network Model of User's Behavior. Int. J. on Information Theories and Applications Vol. 10 Num. 2 (2003) 184–188.

[9] Gorodetski, V., et al.: Agent-based model of Computer Network Security System: A Case Study. In: Proc. of the International Workshop 'Mathematical Methods, Models and Architectures for Computer Network Security', Lecture Notes in Computer Science, Vol. 2052. Springer Verlag (2001) 39-50.

[10] Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Upper Saddle River NJ: Prentice Hall (1995).

[11] Luck, M., McBurney, P., Preist, C.: Agent Technology: Enabling Next Generation Computing. AgentLink (2003).

[12] Haykin S.: Neural Networks: a comprehensive foundation. Upper Saddle River, New Jersey: Prentice Hall (1999).

[13] Wooldridge, M.: An Introduction to Multi-agent Systems. Chichester, England: John Wiley & Sons (2002).

Author's Information

Andrii Yu. Shelestov – *PhD, Senior Researcher, Department of Space Information Technologies and Systems, Space Research Institute of NASU-NSAU, Glushkov Ave 40, Kyiv-187, 03650 Ukraine, e-mail: inform@ikd.kiev.ua.*

АВТОМАТИЧЕСКОЕ ВЫЯВЛЕНИЕ УДАРНЫХ ВОЛН ПО ИЗМЕРЕНИЯМ СПУТНИКА ACE

Андрей Шелестов, Ксения Житомирская, Николай Ильин, Игорь Кременецкий

Abstract: *В работе предлагаются два алгоритма автоматического выявления ударных волн по интенсивности потока ионов на основе измерений спутника ACE, позволяющие определять 89% зафиксированных событий, что в 3 раза превышает точность известного алгоритма. Проведенные эксперименты подтверждают работоспособность предложенных алгоритмов.*

Keywords: *сервисы, обработка данных, космическая погода*

Введение

Основным из известных околоземных проявлений космической погоды, контролируемой солнечной активностью, являются магнитные бури. Влияние магнитных бурь на технику и человека неоспоримо, но их возникновение зависит от целого ряда условий. Согласно современной (господствующей) теории воздействие солнечной активности на Землю осуществляется через истекающий поток солнечной плазмы – «солнечный ветер» (СВ). Наиболее частым из геоэффективных возмущений СВ являются ударные волны (УВ). Ударные волны генерируются при солнечных вспышках, корональных выбросах масс (КВМ), на границах корональных дыр при смешении медленного и высокоскоростного потоков СВ. Генерация УВ происходит в областях солнечной короны, где скорость потока плазмы выше характерной магнитогидродинамической скорости.

Интервал времени между моментом генерации УВ и ее приходом на Землю составляет порядка нескольких дней. Прогнозирование времени прихода и интенсивности УВ поможет принять соответствующие меры по минимизации разрушительных последствий бурь Поэтому дистанционная идентификация УВ в СВ и определение времени ее прихода является одной из самых важных задач краткосрочного прогноза состояния космической погоды околоземного пространства.

Поставленная в работе [1] задача прогноза ударной волны опирается на предположение об известности (или простоте определения) начала ударной волны, но физическое основание для такого предположения и механизм определения самого момента начала не приведены. В данной работе предлагается алгоритм автоматического определения ударных волн и анализируются физические предпосылки его функционирования. Результаты предлагаемого алгоритма более точны, чем полученные в [1], что подтверждается данными независимых наблюдений [2]. Применение предложенного алгоритма при прогнозировании времени прихода УВ позволит существенно повысить точность и достоверность прогноза.

Физические предпосылки дистанционной идентификации ударных волн

Присутствующие во всей Вселенной энергетические частицы не находятся в термодинамическом равновесии. Так как крупномасштабные электрические поля встречаются во Вселенной крайне редко, на протяжении большей части своей жизни заряженные частицы движутся по круговым орбитам под действием одной лишь силы Лоренца. Эта сила позволяет частице сохранять свою энергию постоянной, таким образом, при отсутствии столкновений частица может увеличивать свою энергию только под действием электрического поля.

В 1949 году Ферми сформулировал теорию, согласно которой космические лучи ускоряются при рассеивании заряженных частиц на магнитных облаках, действующих как магнитные зеркала [3]. При встречном столкновении частица приобретает дополнительную энергию.

При ускорении Ферми первого порядка рассматривают два приближающихся зеркала, таким образом частицы колеблются между ними на протяжении многих лет, наращивая энергию при каждом столкновении. Энергия увеличивается пропорционально U/v, где U - скорость облака, v - скорость частицы. При ускорении Ферми второго порядка облака двигаются случайно. Так как столкновения с потерей энергии менее вероятны, то в результате чистый прирост энергии пропорционален U^2/v^2.

Крупномасштабные процессы на Солнце сопровождаются турбулизацией среды, в ходе которой через стохастическое ускорение частицам передается энергия. В результате выбросов корональных масс Солнца формируется высокоскоростной поток плазмы, возбуждающий **ударную волну** (interplanetary shock, shock wave).

Ударная волна является наиболее эффективной конфигурацией, в которой работает эта теория, обеспечивая встречные столкновения для пересекающих ударный фронт частиц.

В ударной волне ускорение частиц происходит двумя способами: дрейфовым и диффузионным. При дрейфовом ускорении на ударном фронте отслеживается движение отдельных частиц в электромагнитном поле ударного фронта, а взаимодействием с флуктуирующими полями пренебрегают. Частицы много раз пересекают ударный фронт и каждый раз увеличивают энергию. Процесс диффузионного ускорения описывается модифицированным уравнением Фоккера-Планка, фактически описывающим диффузии в пространстве скоростей. [4-6]

Постановка задачи выявления ударных волн

Находящийся в точке Лагранжа L1 (~1% расстояния от Земли до Солнца) спутник ACE измеряет интенсивности потоков ионов. Таким образом, до регистрации на спутнике волна проходит порядка 148 млн. км. и, встречая на своем пути другие частицы, ускоряет их по направлению к Земле. Так как степень ускорения частиц пропорциональна скорости волны, то первые из них достигнут Земли приблизительно за два дня до прихода ударной волны. При распространении УВ к Земле интенсивность потоков солнечных энергетичных частиц (СЭЧ) будет расти.

Резкое, не свойственное обычной картине колебаний, нарастание интенсивности СЭЧ во всех диапазонах называют **началом развития ударной волны** (onset) [1].

Результаты измерений для различных каналов спутника ACE, иллюстрирующие процесс развития ударной волны показаны на рис. 1. Из рисунка видно, что однозначно определить момент начала развития ударной волны чрезвычайно сложно даже визуально. Для прогнозирования прихода ударной волны на Землю в реальном времени необходимо разработать алгоритм автоматического определения начала развития ударной волны.

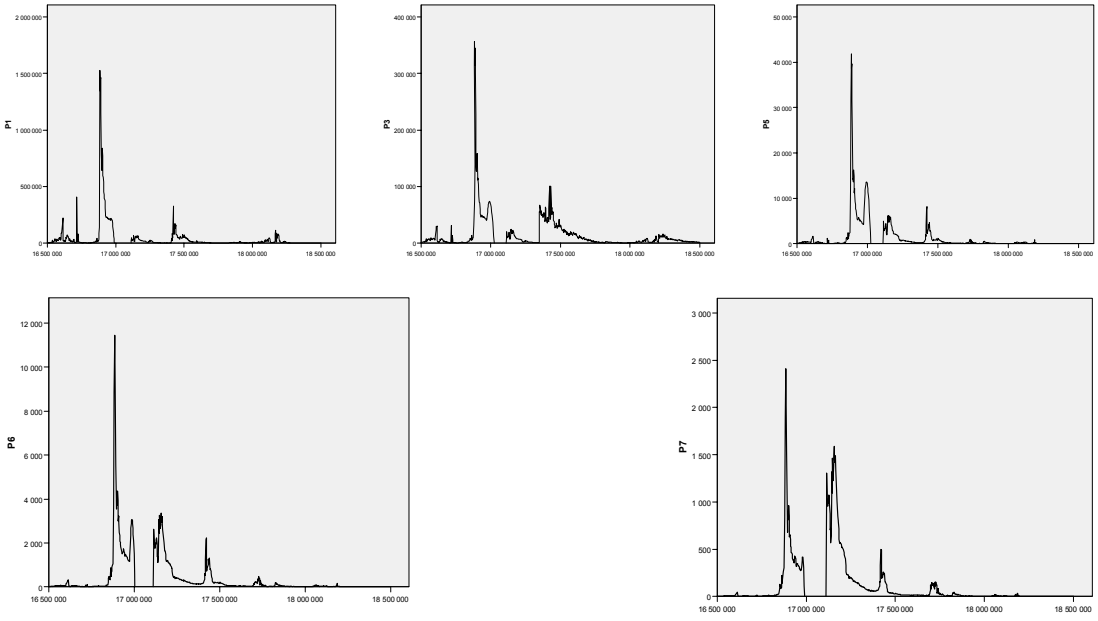


Рис. 1. Развитие ударной волны по измерениям спутника ACE

Несмотря на простоту визуального определения момента начала ударной волны на графиках измерений, разработка алгоритма автоматического детектирования развития ударной волны по данным измерений сопряжена с определенными трудностями. Во-первых, при определении начала развития ударной волны необходимо принимать во внимание измерения по всем информативным каналам спутника, а значит одновременно анализировать несколько числовых рядов. Во-вторых, кривые, построенные по результатам этих измерений являются негладкими и содержат пики различной интенсивности, которые

могут относиться как к одной и той же ударной волне, так и к следующим друг за другом волнам. Поэтому на основе экспериментального анализа данных спутника ACE необходимо также выработать процедуру автоматического определения порогового значения, которое можно использовать для детектирования ударных волн.

Таким образом, в работе ставится задача разработать формальную процедуру (алгоритм) для автоматического анализа измерений спутника ACE, выявления моментов турбулентности и начала развития ударных волн. Результаты работы такой процедуры в дальнейшем можно использовать для прогнозирования времени прибытия ударной волны на Землю в рамках интерактивного сервиса в разрабатываемой под эгидой Национального космического агентства Украины системе "Космическая погода".

Разработанный алгоритм будет реализован в системе прогнозирования прибытия ударной волны в блоке детектирования начала ее развития (рис. 2). Исходные данные для функционирования такой системы в режиме реального времени предоставляются в частности на сервере наблюдения параметров космической погоды http://www.sec.noaa.gov/ftpdir/lists/ace/ace_epam_5m.txt.

Рис. 2. Система прогнозирования прихода ударной волны

Выделение информативных данных

Хотя ударные волны оказывают влияние на все частицы, наиболее существенный прирост интенсивности наблюдается на высокоэнергетичных протонах. Спутник ACE определяет интенсивности по 8 протонным каналам каждые 5 минут.

Для определения информативных каналов необходимо проанализировать поступающие данные. Корреляционный анализ данных показал высокую степень корреляции между близкими по спектру каналами (табл. 1).

Таблица 1. Степень корреляции между отдельными каналами спутника ACE за 1999-2000 годы

	Диапазон частиц (MeV)	P1	P2	P3	P4	P5	P6	P7	P8
P1	0.047-0.066	1	,987	,782	,921	,897	,817	,721	,640
P2	0.066-0.114	,987	1	,794	,941	,919	,825	,710	,618
P3	0.114-0.190	,782	,794	1	,862	,784	,715	,620	,540
P4	0.190-0.310	,921	,941	,862	1	,966	,896	,783	,679
P5	0.310-0.580	,897	,919	,784	,966	1	,964	,873	,778
P6	0.580-1.05	,817	,825	,715	,896	,964	1	,967	,898
P7	1.05-1.89	,721	,710	,620	,783	,873	,967	1	,974
P8	1.89-4.75	,640	,618	,540	,679	,778	,898	,974	1

Как видно из табл. 1, данные каналов P2, P4, P8 целесообразно опустить, поскольку уменьшение пространства задачи до пяти каналов не только не снижает общей информативности данных, но и позволяет существенно сократить объем обрабатываемых данных и как следствие снизить сложность вычислений. При появлении ошибок в основных каналах, данные вспомогательных каналов можно использовать для восстановления утраченной информации. Так, по динамике P2 можно либо спрогнозировать следующее значение P1, либо полностью заменить интенсивности канала P1 на P2.

Предлагаемый алгоритм определения начала ударной волны

Для автоматического определения момента развития ударной волны необходимо выполнить сглаживание данных, устранив высокочастотную составляющую сигнала, а также определить «скорость» нарастания интенсивности частиц (угол наклона сглаженной кривой), учитывая данные всех информативных каналов.

Предлагаемый способ идентификации начала ударной волны основывается на пороговой фильтрации, так как по определению момент начала развития ударной волны характеризуется ростом интенсивности частиц по всем каналам. Для определения начала развития ударной волны предлагается следующий обобщенный алгоритм:

- Построение обобщенного канала и вычисление его скорости;
- Сглаживание обобщенной скорости развития процесса;
- Пороговая фильтрация данных.

Построение общей скорости для каналов может быть произведено несколькими способами. Рассмотрим два варианта алгоритма:

- Вычисление скоростей по отдельным каналам с последующим определением обобщенной скорости;
- Построение обобщенного канала и вычисление скорости изменения интенсивности частиц для этого канала.

Рассмотрим первый случай, предполагающий вычисление скоростей по отдельным каналам с последующим определением обобщенной скорости. Сначала определим скорости по отдельным каналам (обозначенным индексом i)

$$V_i(t) = \frac{dP_i(t)}{dt}, \tag{1}$$

а затем объединим их для получения обобщенной скорости

$$V_M(t) = \prod_{i=1}^{N} V_i(t). \tag{2}$$

В (1)-(2) приняты следующие обозначения: $\frac{dP_i(t)}{dt} = \frac{P_i(t) - P_i(t-\Delta t)}{\Delta t}$ — численно определяемая производная функции интенсивности канала $\mathrm{P_i}$ по времени, $\mathrm{i = 1..N}$; $\mathrm{V_i}(t)$- скорость изменения интенсивности частиц в канале $\mathrm{P_i}$ в момент времени t, $\mathrm{i = 1..N}$; $\mathrm{V_M}(t)$ — общая скорость изменения интенсивности по всем каналам в момент времени t; N - число информативных каналов; Δt — интервал между двумя последовательными измерениями.

Рассмотрим второй случай, состоящий в построении виртуального обобщенного канала и вычислении скорости изменения интенсивности частиц для него. Сформируем обобщенный канал

$$P(t) = \prod_{i=1}^{N} P_i(t), \quad (3)$$

для которого вычислим общую скорость

$$V_C(t) = \frac{dP(t)}{dt}. \quad (4)$$

В (3)-(4) приняты следующие обозначения: $P(t)$ — обобщенная интенсивность в момент времени t, полученная на основе измерений по всем каналам; $\frac{dP(t)}{dt} = \frac{P(t) - P(t-\Delta t)}{\Delta t}$ численно определяемая производная общей интенсивности по времени; $V_C(t)$ — общая скорость изменения интенсивности; N - число информативных каналов.

Формализация алгоритма и сравнительный анализ вариантов модификаций

Помимо описанных выше данных спутник ACE также измеряет скорость и плотность солнечного ветра. Эту информацию можно использовать для верификации получаемых результатов определения ударной волны, поскольку пик ударной волны соответствует резкому изменению плотности и/или скорости солнечного ветра. На основе информации о параметрах солнечного ветра за 2000 год [7] сформируем список событий, которые должны быть определены детектором как ударные волны.

Сначала вычислим обобщенные скорости $V_M(t)$ по формуле (2) и $V_C(t)$ по формуле (4) для всех моментов измерений t. Для снижения влияния случайных колебаний интенсивностей частиц выполним сглаживание полученных значений во временном окне, содержащем данные k последовательных измерений

$$\overline{V}_M(t) = \frac{1}{k+1} \sum_{i=0}^{k} V_M(t + i\Delta t), \quad (5)$$

$$\overline{V}_C(t) = \frac{1}{k+1} \sum_{i=0}^{k} V_C(t + i\Delta t). \quad (6)$$

Размер окна (k=4) был определен экспериментально: меньший размер не обеспечивает эффективного сглаживания данных, а при большем размере окна может быть потеряна наиболее важная информация об экстремальных значениях измеряемых величин.

Для каждого из подходов определим числовые значения порогов T_C и T_M, которые будут использоваться для последующей фильтрации. В первом приближении значение порога предлагается получать по методу минимакса на основе вычисления минимального за рассматриваемый период d значения максимальной общей скорости в дни с выраженной турбулентностью значений интенсивности солнечного ветра:

$$L_M = \min_d \max_{t \in D} \overline{V}_M(t), \quad (7)$$

$$L_C = \min_d \max_{t \in D} \overline{V}_C(t), \quad (8)$$

где d — рассматриваемый диапазон значений дат, D — множество дней с выраженной турбулентностью интенсивности солнечного ветра.

Выполним пороговую фильтрацию с использованием вычисленных по формулам (7) и (8) пороговых значений. Пусть Ω_M и Ω_C — множества значений скоростей, вычисленных по формулам (2) и (4) и отфильтрованных с использованием пороговых значений (7) и (8) соответственно:

$$\Omega_M = \left\{\overline{V}_M(t) : \overline{V}_M(t) \geq L_M\right\}, \quad (9)$$

$$\Omega_C = \left\{\overline{V}_C(t) : \overline{V}_C(t) \geq L_C\right\}. \quad (10)$$

Сгруппируем данные по степени временной близости. На основании статистического анализа исторических данных измерений и архивных данных [7], [2] в данной работе принято предположение о параметрах ударной волны – отфильтрованные измерения, отвечающие скачкам интенсивности, между которыми прошло не более 12 часов, относятся к одной ударной волне.

Результаты экспериментов

Для определения корректности применения предложенных в предыдущем разделе вариантов алгоритма и определения числовых значений порогов был проведен анализ данных за 2000 год, в результате которого были построены предполагаемые периоды ударных волн. Экспериментальное исследование построено таким образом, чтобы на основе известных периодов ударных волн получить значения ошибок первого и второго рода для двух вариантов алгоритма.

Из 60 реальных дат ударных волн, по критерию V_C были определены как относящиеся к ударным волнам 53 даты, что соответствует ошибке 1-го рода равной 0.117, а по V_M – 44 даты (0.267).

При сравнении полученных результатов с наблюдениями по солнечному ветру, оба варианта алгоритма дали одинаковое значение ошибки 2-го рода (ложное оповещение о приближении ударной волны). Ошибка 1-го рода для второго варианта составляет 0.117, что существенно меньше, чем 0.267 для первого варианта, что позволяет говорить о предпочтительности использования обобщенного канала.

Выводы

В данной статье предложены два варианта алгоритма определения начала развития ударных волн.

Результаты работы алгоритма позволяют не только существенно повысить эффективность предсказания прихода ударной волны, но и рассматривать при обучении практически все зафиксированные ударные волны. Начало удалось зафиксировать в 89% ударных волн, что в 3 раза больше чем в известном ранее алгоритме [1]. Одним из путей повышения точности определения ударных волн является введение коррекции значений порогов в режиме реального времени.

Литература

[1] J. Vandegriff, K. Wagstaff, G. Ho, J. Plauger Forecasting space weather: Predicting interplanetary shocks using neural networks, *Advances in Space Research* 36 (2005) p. 2323–2327

[2] Списки космических бурь за 1997-2006 г.г. (ИЗМИРАН)

http://helios.izmiran.troitsk.ru/cosray/events.htm

[3] Fermi E. On the origin of the cosmic radiation // Phys. Rev. 1949. V. 75. P. 1169-1174.

[4] Blandford R.D., Eichler D. Particle acceleration at astrophysical shocks: a theory of cosmic ray origin // Phys. Rep. 1987. V. 154. 1-75.

[5] Jones F.C., Ellison D.C. The plasma physics of shock acceleration // Space Sci. Rev. 1991. V. 58 P. 259-346.

[6] Kirk J.G. Particle acceleration // Plasma Astrophysics / Ed. By J.G. Kirk, D.B. Melrose, E.R. Priest. – Berlin: Springer-Verlag, 1994. P. 225-314.

[7] Интерактивный архив данных по солнечному ветру

http://www.srl.caltech.edu/ACE/ASC/afs/SWEPAMdata.html

Информация об авторах

Шелестов Андрей Юрьевич - *Институт космических исследований НАН и НКА Украины, кандидат технических наук, старший научный сотрудник, докторант; проспект Академика Глушкова 40, 03680 Киев, Украина; e-mail: inform@ikd.kiev.ua*

Житомирская Ксения Геннадиевна - *Институт космических исследований НАН и НКА Украины, инженер-программист 1 категории; проспект Академика Глушкова 40, 03680 Киев, Украина; e-mail: ksu.zhytomirsky@gmail.com*

Ильин Николай Иванович - *Институт космических исследований НАН и НКА Украины, инженер-программист 1 категории; проспект Академика Глушкова 40, 03680 Киев, Украина;*

Кременецкий Игорь Алексеевич - *Институт космических исследований НАН и НКА Украины, кандидат физико-математических наук, научный сотрудник; проспект Академика Глушкова 40, 03680 Киев, Украина.*

SAFETY POLICY PROBLEMS OF CLUSTER SUPERCOMPUTERS

Andrey Golovinskiy, Sergey Ryabchun, Anatoliy Yakuba

Abstract*: The paper describes the problems of safe management and safe work of the supercomputer protected both from purposeful attacks from the outside, and from results of wrong activity of its usual users. The paper may be useful for those scientists and engineers that are practically engaged in a cluster supercomputer systems design, integration and services.*

Keywords*: supercomputer, cluster, computer system security policy, virtualization.*

ACM Classification Keywords*: C.1.4 Parallel Architectures. C.2.4 Distributed systems, D.4.7 Organization and Design*

Introduction

The powerful computer center is a titbit for the malefactor. The information about tens users (scientists, programmers) is usually stored into system databases, giving links to their personal computers with the valuable scientific and technical information or the developed software. The disk files of a supercomputer are stored the confidential system data, many personal programs of users. Original development of supercomputer founders also can be the object of interest.

Besides the supercomputer equipped with the broadband Internet channel, can be convenient jumping-off place for carrying out of attacks to other servers of a local network and the internet.

The basic threats to computing system (CS). We shall allocate three basic threats CS:

- **Threat of disclosing,** consists that the information which did not intend for a wide circulation, becomes known to a uncertain circle of persons. It is most known of threats.
- **Threat of integrity,** consists that as a result of some activity of users (they are not necessary malefactors) the stored information can be deformed, for example, in the environment of data transmission or in a place of its storage.
- **Threat of refusal of services.** This threat consists that the malefactor certain actions can cause faults in work of some system services.

The organization of supercomputer safety

The non-authorized access from which it is necessary to protect CS, it is possible to divide into two unequal parts.

- Any non-authorized access from the outside,
- Non-authorized access of the authorized supercomputer user.

The common sense prompts, that with a view of safety with an external world it is necessary to limit interaction by an entrance gateway of a supercomputer, with a minimum of the interface of interaction, for example protocols ***ssh***, ***https*** (the protocol ***http*** is possible as a variant of access to the general information only in a read-only mode). Besides external access to some technical services, to services ***DNS, NTP*** should be open. Ports of other services should be filtered.

Open services ***ssh, https, DNS*** can be exposed to external attacks, measures on reflection of such possible attacks therefore should be undertaken.

For the analysis vulnerable points inside supercomputer we shall divide into such groups:

1. The active technical equipment - managed switches, devices of an uninterrupted supply (UPS), devices of the removed management of servers (through controllers with realization of protocol IPMI) - usually is in service nets. These nets should be accessible only to system administrators.

Possible variant of the decision is as follows: allocation of such equipment in a separate virtual net in which there is an access only from a gateway, access to which, in turn, is adjusted by the net screen (Firewall)

2. Service of identification of users LDAP should have the reliable password, this password is accessible in some system scripts.

3. Scripts of administration managerial control.

4. A system of supercomputer resources handling. Its own means of protection should be involved in it.

5. Linux kernel.

6. Open ports of global file system Lustre [1].

7. Interconnect with MPI-protocol at cluster nodes.

All these create big amount of problems concerned to supercomputer safety. In everyone concrete supercomputer there is a specificity, but the common feature is potentially a plenty of gaps in protection and complexities of their overcoming.

Protection of workplaces of system administrators

Working computers of system administrators - a special class a supercomputer component, special in many senses. First, by initial development of structure of a supercomputer these computers frequently appear outside of a field of vision and have no enough the thought over protection. Second, they can keep strictly confidential data about management of the supercomputer – tuning, internal reports on safety, working notes, backup copies of software, etc. Already only this transfer demands, that to a safety of workplaces of administrators has been paid not smaller attention, than to safety of the open services.

The natural decision - to have workplaces of system administrators into physically allocated subnet with two points of interaction, one of them is an entrance supercomputer gateway, and the second - a gateway delivering the internet in the corporate local computer net. Thus in a gateway it is not necessary to create a separate operating mode of administrative computers, they should work with the rights of any external computer from the internet. Besides access from within supercomputer in the administrator subnet should be allowed only to system administrators. As if to access from the outside in this net it should be completely closed for all.

One of possible options to decide a problem of storage of working files is the organization of the X-net of terminals for work of administrators. With such an option all working materials will lay on one terminal - server on which it is easier to provide their reliable and safe storage.

Internal supercomputer nets

The organization of a net. Actually, protection of a gateway though is necessary, but rather conditional. At enough big base of users, a part from them it is necessary, even once, will begin a session of connection with supercomputer through a computer which is infected either virus, or the Trojan program which collect passwords and send them in hypothetical outside database. Therefore it is enough to malefactor to address to such database to receive the information for an input inside of a supercomputer. Therefore protection of a supercomputer is necessary for projecting in view of that a quantity of malefactors can penetrate into.

System of storage of backup copies. In the chosen model of safety the unique variant of correction of consequences of breaking is full reinstallation of operational systems on all servers of a supercomputer. It is labour-consuming procedure and it is practically equivalent to construction of a supercomputer from nothing.

To minimize consequences, it is expedient to allocate a separate server on which backup copies of all supercomputer systems will be stored. To secure this server against breaking, it is necessary to close any input to it from the net, to leave only an input from the console. Such storehouse is useful and at various failures.

Construction of protection for service of Grid-calculations. Specificity of Grid tasks is that the executed code and the data can come from any point of the world and from the unknown user. Therefore it is necessary to provide performance of such problem in the allocated container, without interaction with other components of a complex. In this case the container is set of the units allocated for the decision of one task.

As the common requirements it is possible to result the following in the container:

1. The condition of the container after performance of a task should correspond precisely to a condition up to it.
2. Breaking unit should not entail consequences for supercomputer.
3. Interaction of the user and its task with supercomputer is limited to units of the container and a gateway.

The first requirement is rather easy for resolving, if root file system to give accessible only for reading and to overload unit on the termination of a task with the subsequent deleting contents of a local disk.

The second requirement to provide much more difficultly. At the superuser on unit it is a lot of opportunities, access to base of users, wide access in the general file system, etc.

The third requirement is rather easily feasible, for example, with the help of functionality of virtual nets (VLAN) on managed switches.

The basic minus of the tendency of closing *of all* holes and passes in system is excessive complication of system, it inevitably entails occurrence of new gaps in protection. Technology which presumes to execute all these requirements with comprehensible productivity and with rather small complication of system, is ***virtualization***.

Virtual supercomputer

Supercomputer can be subjected to attack outside, but not less threats attack from within represents. Very much frequently on computing supercomputer programs are started, to check up which on safety it is not obviously possible. These programs are created literally on the move, compiled and sent in turn on performance. To make audit of a code of such programs it is unreal - the amount of these programs, their often updating and round-the-clock work supercomputer will make such attempt impossible. Therefore always there is a danger, that any of the started programs will contain a nocuous code.

Danger especially grows if to take into account, that to the author of a code "subtleties" of the environment of execution can be known. The environment of execution is here enough static, that is, as a rule, the used equipment imposes restriction on the system software that entails rare enough updating and, accordingly, means presence of some number enough for a long time found out and already corrected gaps in protection of which the malefactor can always take advantage for reception of the partial or full control over resources. Thus the purposes of attack can be anyone - access to the confidential data, infringement of integrity of the data, refusal in service, attack on external in relation to supercomputer objects and so on.

Clearly, that one of the purposes of construction of a policy of supercomputer safety should be protection of the supercomputer against such attacks, that is, from attacks from within. Complicates the decision of a task in view necessity to take into account a lot of additional conditions:

- For a parallel task the allocated resources are uniform object and toughening of protection of each separate node can lead to failure of any task;
- The supercomputer is the multiuser and multitask system.

Function chroot* as a variant of the decision of a problem.** Function ***chroot allows system to start process, using the certain catalogue as root. Thus it is possible to limit access to strictly certain data for process. And not only to limit, process can receive completely other environment of execution, for example, 32-bit ALTLinux [2] environment that root system is 64-bit CentOS [3] environment.

Very attractively, but, unfortunately, it works for a single computer. For supercomputer the application of the operator ***chroot*** only will complicate a problem to the malefactor, but full problem solving in a supercomputer cannot be reached.

The reason consists that the computing node is not an independent element - it is closely connected to other nodes which are included in a resource of a task, and also through a management system with all other supercomputer nodes. Thus. It is possible both net attack, and attempt to receive the full control over any node through a probable gap in a kernel.

Clearly, that use of functionality ***chroot*** for the decision of specific problems is allowable, and use as means of protection not only will not give necessary effect, but even harmful, as can give the manager of a supercomputer feeling of false security.

Nevertheless, the problem of creation of system of safety is, it is real also it is necessary to solve. By what means? From our point of view protection of separate elements of an infrastructure cannot be optimum, always there will be a unprotected element through which attack is possible. The most optimum is allocation for a problem of the user virtual supercomputer inside which the user is the full owner with some restrictions, namely, with the minimal administrative opportunities.

Thus, the supercomputer turns to a set from one or more virtual supercomputers with one problem and one user in everyone, and process of start of a problem of turn becomes complicated item of creation virtual supercomputer. Itself virtual supercomputer it will be submitted as a set of virtual computing nodes.

Requirements to such virtual supercomputer are simple enough - *the allocated net address space* and *the allocated file system*.

Let's try to formulate requirements to system virtualization which we shall use on computing nodes.

High efficiency. Computing supercomputer, real or virtual, has the main function - quickly to count, therefore an obligatory overhead charge on virtualization should be minimal, in an ideal case this charge should aspire to zero.

Opportunity of direct use of the allocated equipment. As the computing environment in supercomputer are frequently used the various high-efficiency communication equipment, for example, in the SCIT - 1 and the SCIT - 3 supercomputers it is InfiniBand, and in SCIT - 2 SCI [4] and if the virtual node will not have to it direct access falling of productivity of all virtual supercomputer will be rather essential enough.

Virtualization types. *Emulation of the equipment* - in host-system is created the virtual machine modelling some hardware platform (fig. 1). As each command of the processor should be simulated on a real platform productivity can fall in tens and even hundreds times.

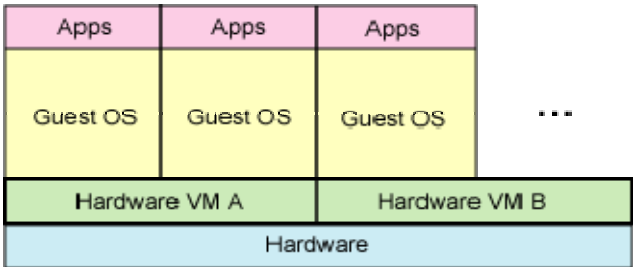


Fig. 1. Emulation of the equipment

Full virtualization - uses a program layer between guest operating systems and the equipment, *hypervisor* (fig. 2). Losses of productivity low and it can be started the unmodified guest systems, but also there is very short list of really supported equipment.

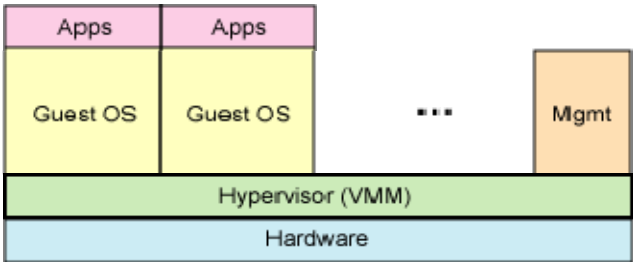


Fig. 2. Full *virtualization*

Paravirtualization - this method is similar with full *virtualization*, difference that hypervisor belongs to kernel host-systems (fig. 3). This method gives the productivity close to unvirtualized system, use of the modified guest systems however demands.

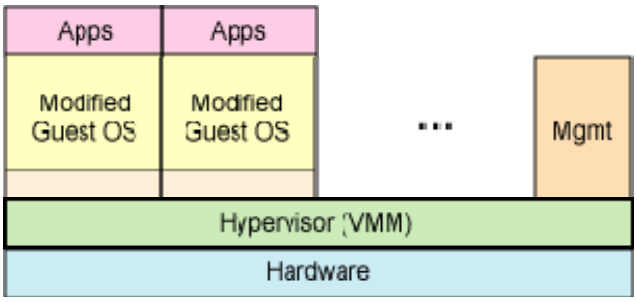


Fig. 3. Paravirtualization

A level of operating system virtualization - actually it is simple isolation of the containers, allowing to receive the productivity practically equal to initial productivity (fig. 4).

The systems realizing p*aravirtualization* and a level of operating system *virtualization* concern to examined real candidates. However, it is necessary to refuse from paravirtualization too, as use in supercomputers enough the rare equipment generates absence of support of this equipment by the hypervisor.

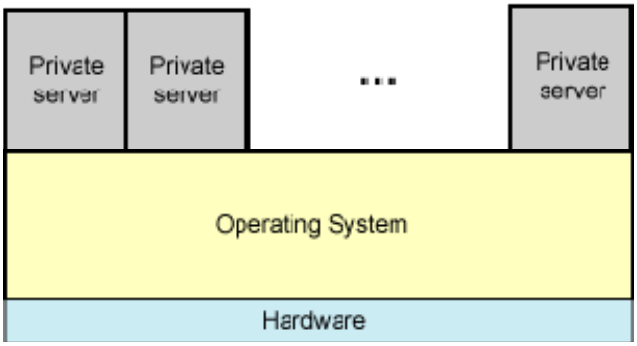


Fig. 4. A level of OS virtualization

Thus, only a level of operating system *virtualization is* suitable to our conditions. Though the operator ***chroot*** on the functionality to this type *virtualization* also concerns, but, unfortunately, it can *virtualize* only file system, it is required to us much more.

Examples of systems virtualization

OpenVZ - realization of technology of a virtual level of operating system on Linux kernel. **OpenVZ** allows to start on one physical device some the isolated environments named VPS (Virtual Private Server) or VE (Virtual Environments) [5].

OpenVZ gives excellent productivity - falling does not exceed 1-2 % on modern systems, scalability - on one physical device can be started some hundreds virtual environments, dynamic resource management is realized, administration differs simplicity. **OpenVZ** also supports migration of virtual environments that allows to transfer hurriedly practically the virtual environment from one physical server on another, and accordingly, allows to build various scripts of use (unfortunately, this opportunity in кластерном a supercomputer environment can be used with big enough clauses as restriction is imposed with drivers of used interconnect).

Linux-VServer - realization of technology of *virtualization* a level of operating system [6]. As well as **OpenVZ**, allows to start on one physical server a little isolated VPS. But as against **OpenVZ**, uses the concept of the expanded functionality ***chroot*** with expansion on isolation of the processor, memory and net resources. Does not support migration and has more simple net support that entails additional complexities in administration.

Clearly, that utilization of this method will not give full safety also, a gap is the opportunity of use for attack of the computer net. Nevertheless, this method gives high enough level of security.

Conclusion

In the near future we are going to carry out experiments on creation virtual supercomputers with application **Linux-VServer** and **OpenVZ**, and, in case of successful result, we shall introduce one of them in the standard circuit of use.

Further we plan to finish working versions MPI for exception of use of direct access of process to low level interfaces of the computer net. In case of the successful decision of this problem we can receive completely safe virtual supercomputer.

Bibliography

[1] http://clusterfs.com/

[2] http://altlinux.org/

[3] http://centos.org/

[4] A.Golovinskiy. S.Ryabchun, A.Yakuba. Cluster supercomputer architecture. In: Proceedings of the XII-th Int.Conf. “Knowledge-Dialogue-Solution”- Varna, 2006.

[5] Linux. An overview of virtualization methods, architectures, and implementations. http://www-128.ibm.com/developerworks/linux/library/l-linuxvirt

[6] Linux V-Server Overview. http://linux-vserver.org/Overview

Authors' Information

Andrey L. Golovinskiy *- email: tikus@ukr.net*

Sergey G. Ryabchun *- email: serge.ryabchun@gmail.com*

Anatoliy A. Yakuba *- email: ayacuba@gmail.com*

Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova, 40, Kiev, 03680 MCP, Ukraine;

PARAMETRIC IDENTIFICATION AND DIAGNOSIS OF INTEGRATED NAVIGATION SYSTEMS IN BENCH TEST PROCESS

Ilya Prokoshev, Alexander Chernodarov

Abstract*: Growth of complexity and functional importance of integrated navigation systems (INS) leads to high losses at the equipment refusals. The paper is devoted to the INS diagnosis system development, allowing identifying the cause of malfunction. The proposed solutions permit taking into account any changes in sensors dynamic and accuracy characteristics by means of the appropriate error models coefficients. Under actual conditions of INS operation, the determination of current values of the sensor models and estimation filter parameters rely on identification procedures. The results of full-scale experiments are given, which corroborate the expediency of INS error models parametric identification in bench test process.*

Keywords*: fault detection, integrated navigation systems, state control, sensors, model of errors, parametric identification, supervision, monitoring, fault diagnosis, diagnostic reasoning*

ACM Classification Keywords*: B.8.1 Reliability, Testing, and Fault-Tolerance; J.2 Computer Applications: Physical Sciences and Engineering: Aerospace*

Introduction

Most model-based methods for fault detection and diagnosis rely on the idea of analytical redundancy that is the comparison of the actual behavior of a system to the behavior predicted on the basis of the mathematical system model. Typical model-based fault detection process consists of two steps: residual generation and residual assessing/classification. The decision making is actually a process of classifying the residuals into one of two classes: normal and fault.

Nowadays, the necessity for an inertial support of the operation of integrated navigation systems is considered to be proved. Such a support forms is the basis for the highly maneuverable objects continuous navigational support. However, as for the implementation of the potentialities of inertial navigation systems (INSs), the problem of improving their operational characteristics still remains topical. Among such characteristics which significantly affect the navigational safety we may reckon the INS operational-readiness time, INS accuracy, and INS reliability.

Regarding INSs, traditional approaches [1] to the solution of the above problem rely on the hardware modernization of existing sensors and on the development of new types of sensors such as a gyroscope and an accelerometer. Approaches [1], which involve INS error estimation from data obtained from satellite navigation systems (NSs) and from other external NSs are also deemed to be promising. Furthermore, insufficient attention is given, in our opinion, to the study of the capabilities of combined INSs, built around sensors that are different in the: principle of operation. At the same time, available engineering solutions [2] of such a problem provide the necessary basis in order for studies in this particular field to be conducted.

The evolution of INS relies on improvements both in hardware and in the methods of integrating this hardware. The potentialities of INSs [1] are based on the Kalman filtering technology and on the mathematical INS sensors error models. In order for the INS state to be estimated reliably, the parameters both of models and of an optimal Kalman filter (OKF) must reflect actual measuring processes and noise conditions adequately. Therefore, it is essential that during use of INSs, any changes in noise statistics as well as in dynamic and accuracy sensors characteristics be taken into account. This can be done through the identification and retuning of the appropriate coefficients in an algorithm for data processing.

The potentialities of such a technology make it possible

- to combine dissimilar measurement aids into an integrated structure and to improve the accuracy and reliability of navigational determinations on this basis;
- to implement the mutual support of INSs in the interests of ensuring their integrity;
- to estimate both INS errors and sensor errors from indirect measurements and through the use of correlations;
- to form procedures for the monitoring, diagnosis, and control of the INS technical condition.

However, the effectiveness of OKF application as a kernel of INSs essentially depends on the goodness of fit of the mathematical INS error models and sensor errors to actual measuring processes.

Problem Statement

Generally, the structure of navigating system can be presented in the form of three modules, namely: information sensors, information signal converters from the analogue form into digital and digital processing devices (see figure 1).

The experience of inertial navigation systems development shows that the intrinsic error of these units defining their functional reliability is the random parametric drift called by dynamically-tuned gyros, interface electronic cards, control cards and couplers. The given task solution is impossible without more profound analysis of occurrence reasons and influence of design and technological parameters on values and stability of random drift.

According to stated, the research of the factors influential in involuntary drift of system and creation of the effective diagnostic technique permitting to estimate current technical condition of INS is the actual task.

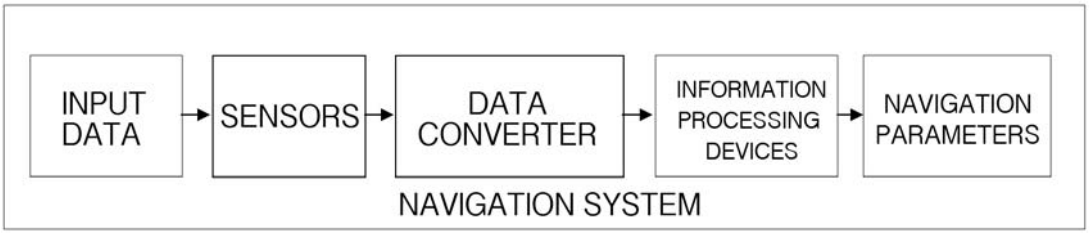


Fig. 1 The structure of typical navigation system

The main work purpose is development of algorithms for the INS diagnosis, permitting to reveal reasons of refusals and faults on the data on the basis of structural adapting and navigation model parameters identification.

The offered solution technology includes the following stages:

- the structural adapting of the INS equations in view of the detected disorder and model defect in parametric type;
- retrospective estimation of the extended state INS error vector, originating because of defects;
- correlation processing of the received estimations of errors;
- solution of the algebraic equations on parameters, approximating correlation function and included in diagnostic model;
- INS state handle in view of the current state of meters, namely - retargeting of parameters of error model and INS working capacity restoring.

Given technology will allow solving the following problems:

- optimization malfunctions search strategy;
- separate system units technical condition estimation.

According to the purpose of work it is possible to solve the following research problems:

- the statistical analysis of INS units parameters accuracy not meeting the quality specifications requirements;
- refusals database development of INS interconnected units not past a trial stages;
- open architecture development for processing information from various data sources;
- development realizing automated information capturing for its subsequent processing.
- Parametric Identification of the Error Models of INS Sensors

The full analysis of various methods has led to expediency of application of complex monitoring systems which use different by the physical nature research methods that, in turn, will allow excluding lacks of one method and use advantages of other methods to realize thus a principle of "redundancy" increasing reliability of INS systems.

The integration of navigation systems relies on a priori known models of the errors of sensors, namely, of accelerometers, gyroscopes, pseudorange sensors, and so on. However, when INSs are in use, the sensors characteristics undergo certain changes. This has necessitated taking into account these changes through the models of errors of the sensors. Thus, the problem of identifying the models of sensor errors parameters arises. The above problem can be solved on the basis of correlative processing of the estimates of sensor errors both in real time and in the postprocessing INS diagnosis.

The equations describing INS functioning in an operating mode can be represented in a general view

$$\dot{y}(t) = F(y,t); \tag{1}$$

$$\dot{y}_p(t) = F(y_p,t) + G(t)\xi(t), \tag{2}$$

where $y(t)$ – n-dimensional state parameter vector of ideal (undisturbed) INS;

$y_p(t)$ – n-dimensional state parameter vector of real (disturbed) INS;

$F(y_p, t)$ – vector function;

$\xi(t)$ – r-dimensional INS disturbance vector;

$G(t)$ – link coefficient matrix of $\xi(t)$ vector with $y_p(t)$ vector ($n \times q$ matrix of variable coefficients describing the INS sensors noise intensity).

The following INS error equation could be put in conformity to the correlations (1) and (2):

$$x(t) = A(t)x(t) + G(t)\xi(t); \tag{3}$$

where $x(t)=y_p(t)-y(t)$ - INS error vector (n-dimensional INS disturbed vector parameters deviation from undisturbed INS vector parameters);

$$A(t) = \frac{\partial F(y,t)}{\partial y}\bigg| y = y_{P(i)} \text{ - } n \times n \text{ partial derivative matrix.} \tag{4}$$

In the onboard implementation of the models of INS errors, it is deemed possible to have an approximate description of the gyro random drift and the accelerometer random displacement as the Markov Gaussian first-order process

$$\Delta \dot{\mu} = -\frac{1}{\tau_\mu}\Delta\mu + \xi\sigma_\mu\sqrt{\frac{2}{\tau_\mu}} \tag{5}$$

with the exponential correlation function

$$R_\mu(t) = \sigma^2{}_\mu e^{-\alpha_\mu|t|}, \tag{6}$$

where t – running time, $\alpha_\mu = 1/\tau_\mu$; τ_μ is the correlation time; $R(0) = \sigma^2{}_\mu$ is the error variance; σ – mean-square deviation of random error; $\xi \in N(0,1)$; μ – sensor related index ($\mu = a$ – accelerometer; $\mu = \omega$ – gyroscope).

Taking into account $A_\mu = \tau_\mu{}^{-1}$; $G_\mu = \sigma_\mu\sqrt{2/\tau_\mu}$ the equation (5) could be embedded in general structure INS error equation (3).

In the equations, coefficients A_μ, G_μ are a priori defined for sensor nonfault states. The change in the technical condition of sensors during use of INSs has an influence both on the parameters and on the structure of the models of errors. Therefore, the need arises for refinement of the models during bench tests. For this purpose, we propose that the technology of structural parametric identification of the models should be used. Such a technology is based on the requirement that the base models of sensor errors be in agreement with the results of correlative processing of estimates in real time. Provision is made for the extension of the sensor errors models, which is adequate to the form of correlation functions when the postprocessing of bench test data is performed.

In relations (1) and (2), the quantity α_μ is the parameter that is to be identified. In this case, the problem can be reduced to the finding of the α_μ value, which minimizes the quadratic function

$$\hat{\alpha}_\mu = \operatorname{argmin}\sum_{j=1}^{N}(\hat{R}_{\mu j} - \sigma_\mu^2\, \mathrm{e}^{-\alpha_\mu\tau_j})^2, \tag{8}$$

where $\hat{R}_{\mu j}$ is the correlation function that is determined from experimental data in the following way:

$$\hat{R}_k = \frac{1}{N}\sum_{i=k+1}^{N+k} \overset{\circ}{x}_i \; \overset{\circ}{x}_{i-k}, \; k = \overline{0, N}; \tag{9}$$

$$\overset{\circ}{x}_i = x_i - m_x; \quad m_x = \frac{1}{N}\sum_{i=1}^{N} x_i \ ,$$

where $x_i = x(t_i)$ – estimated error of corresponding sensor; N –number of retrospective counts;

$$\tau_j = j\Delta t_i;\ \Delta t_i = t_i - t_{i-1},$$

t_i – discrete instants of time.

As for NSs, the present state of their hardware support and mathematical-and-software support makes it possible to extend the field of application of the methods of integrated data processing by "entrusting" these methods with the solution of unconventional problems. Among such problems we may reckon monitoring, diagnosis, identification, and estimation of the NS technical condition from bench tests data.

A traditional estimation scheme is an open-loop one intended to compensate for the estimates of INS errors; this scheme is shown in fig. 2, where the following notation is introduced: y_{DS} is the vector of parameters that are reckoned by the INS; y_{EIS} is the vector of parameters that are reckoned by the external information sensors; z is the vector of observations; $\hat{x}$ is the vector of estimates of INS errors, $\omega(t)$ - dynamic system disturbance vector; $\vartheta(t)$ - disturbance vector in observation channel.

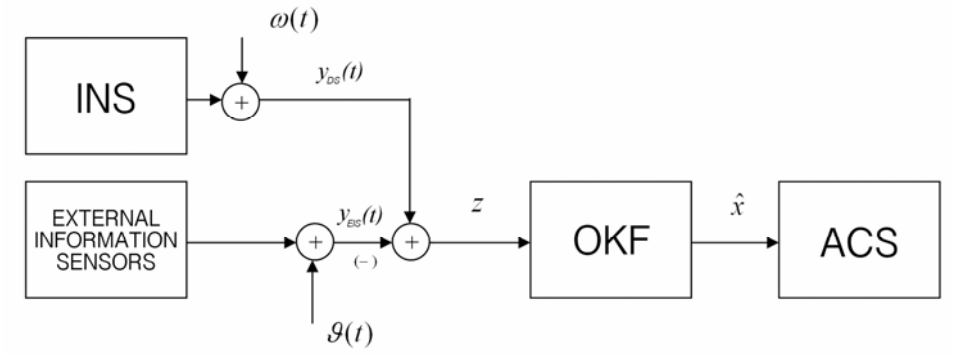


Fig. 2 Open-loop scheme for the damping of INS errors

Using inertial navigation signals and external navigation sensors observations, pseudoranges and velocities measurements the INS error vector estimation could be performed in the following way:

$$z_k = [\varphi\lambda\overline{V}]^T{}_{INS} - [\varphi\lambda\overline{V}]^T{}_{EIS} \tag{7}$$

In the equation (7) φ, λ – geographic latitude and longitude location of moving object; $\overline{V}$ – relative velocity vector of moving object.

Functional monitoring of integrated NSs relies on the technique of channel-wise (element-wise) processing of the vector $z_i = \{z_1,...,z_l\}$ of observations. Based on such a technique, it is apparently possible to check an NS by means of generalized parameters that characterize the state of each of the l measuring channels. For instance, in order to check the j-th measuring channel, use can be made of the normalized residual $\beta_j = \nu_j / \alpha_j$, where α_j is a scaling parameter; $j = \overline{1,l}$; l is the dimensionality of the vector z_i of observations. When observations are processed in "forward" time, the residual ν_j is the difference $\nu_j = z_j - \hat{z}_j$ in value between the actual observation z_j and the predicted observation $\hat{z}_j = H_j \hat{m}_j$, where m_j, $\hat{x}_{i/i}$ is the estimate of the NS error vector x_i at the i-th step after the j-th component and the whole of the vector z_i of observations are processed, respectively; H_j is the row vector of coupling coefficients. Statistical properties of the above-mentioned residuals can be used for the construction of decision rules.

As is known [4] in the absence of discrepancy between the predicted and real observation, the square of the normalized residual β_j^2 is distributed as χ^2, and the quotient of the actual variance $\hat{\alpha}_j^2$ and the predicted variance α_j^2 has the ϑ^2 distribution.

For the distributions in question, the mathematical expectation and variance have tabulated values. These can be used for the formation of tolerances and for the classification of the types of technical condition of an integrated navigation system [5], i.e., of the good condition, operable condition, etc.

Necessary conditions for the good state of the integrated navigation system in reference to the *j*-th component z_i of the vector of observations follow from the properties of the residual ν_j, and they have the form

$$\nu_j \in N(0,\alpha^2); \qquad \beta_j^2 = \nu_j^2 / \alpha_j^2 \in \chi^2(1,2); \qquad F_j = \hat{\alpha}_j^2 / \alpha_j^2 \in \vartheta(a,b),$$

where $\hat{\alpha}_j^2$ by is a true value of the variance of the *j*-th residual, computed on a moving time interval; *a, b* are the tabulated values of the mathematical expectation and variance for the ϑ^2 distribution.

Using the "three-sigma rule" as well as the properties of the χ^2 and ϑ^2 distributions, one can form the tolerances γ_j^2 and η_j^2 respectively on the good and operable condition of the integrated NS in reference to the *j*-th vector of observations channel , i.e.,

$$\beta_j^2 \le \gamma^2 = 1 + 3\sqrt{2} \approx 5.2; \qquad F_j \le \eta^2 = a + 3\sqrt{2b}.$$

The parameter β_j^2 is formed using the current residual and it reflects the current status of *j*-th channel of the vector of observations. If it is out of the tolerance γ^2 , this fact may be associated both with outliers and with failures. The parameter F_j is the quotient of the actual and predicted variance of the residual. It is formed over an averaged range of values of the residual on a moving time interval. Therefore, if it is out of the tolerance η^2 , this fact may be associated with a gradual failure.

The above method intended for the estimation and functional monitoring permits one to establish only the fact that there is a discrepancy between the output signals of the NSs being united, and this fact manifests itself by means of the appropriate components of the vector of the residuals ν_i. Because of this, in order for the diagnosis to be performed, it is apparently expedient to make use of generalized parameters such that the discrepancy would be ascertained for each component of the state vector of an integrated navigation system. In what follows, we show that in order to localize a trouble for the depth of a sensor, namely, of an accelerometer, a gyroscope, it is possible to use some of the combinations of estimates that were obtained in the processing of observations in "forward" and "backward" time.

Analysis of the Results of Studies

The INS-2000 integrated inertial satellite navigation system [3], developed by the RDC (Ramenskoye) has been the object of experimental studies. One of the experiments has been carried out using a bench set of the INS-2000 system, mounted on a geodetically tied-in rotary table. The rotary table was considered as a reference base intended for determination of the actual phase path. The Poisson algorithm for the reckoning of the geographical position from direction cosines and of the projections of the vector of relative velocity on the axes of an inertial measurement unit (IMU) is a navigational kernel of the INS. The basic state vector was comprised of 18 parameters, namely: errors of the IMU angular position; errors in the reckoning of the components of the vector of relative velocity in the direction of the IMU axes; errors in the reckoning of direction cosines; IMU angular drifts and displacements of accelerometers. The results of INS state estimation are shown on fig. 3

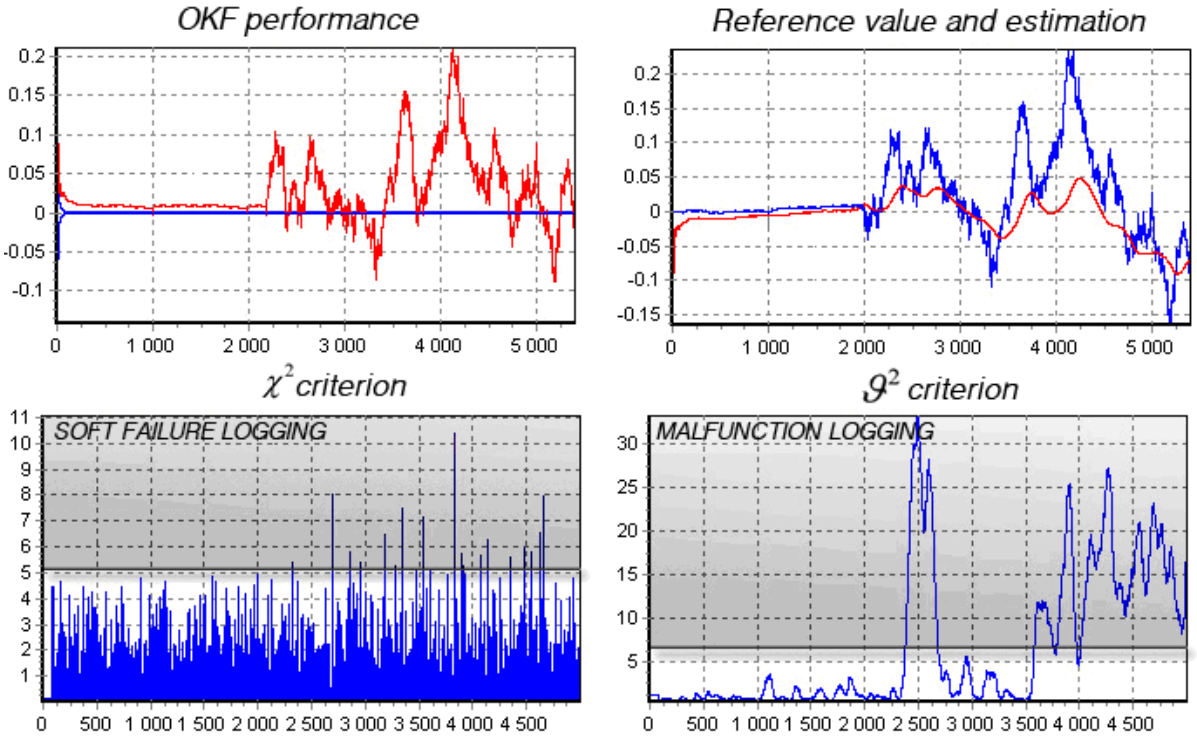


Fig. 3 Results INS state estimation and malfunction logging

Conclusion

In the paper presented here, the authors draw your attention to the importance of systems approach to the construction of mathematical and software support for integrated navigation systems (INSs). Such an approach enables us to combine the capabilities of algorithmic and hardware means intended to improve the accuracy and reliability of INSs. The algorithms considered can form a basis in the construction of a unified technological process, meant for estimation, identification, and control of the INS state. The unified technological process implementation is of great importance in the creation of INSs that provide safety in the use of moving objects.

Bibliography

1. **Schmidt G.T.** INS/GPS Technology Trends. – In: Advances in Navigation Sensors and Integration Technology. RTO Lecture series 232 (2004). Preprints, pp. 1/1 – 1/16.
2. **Fitzgerald R.J.** Divergence of the Kalman filter // IEEE Trans. on Automatic Control. 1971.Vol.16. № 6.– P. 736-747.
3. **Titterton D.N., Weston J.L.** Strapdown Inertial Navigation Technology, Second Edition. Progress in Astronautics and Aeronautics Series, Vol. 207, 2004, 574 pp.
4. **Kailath T.** An innovations approach to least squares estimation. Part 1: Linear filtering in additive white noise // IEEE Trans. on Automatic Control, 1968, Vol. 13, № 6. pp. 646–655.
5. **Sindeev I.M.** (editor). Diagnosis and prediction of the technological condition of aircraft equipment [In Russian].- M: Transport, 1984.

Authors' Information

Ilya V. Prokoshev *- A.A.Baikov Institute of Metallurgy and Materials Science of Russian Academy of Sciences, leading engineer, candidate of science, P.O.Box: 119991 GSP-1, 49, Leninskii Prospect, Moscow, Russia, +7(495)1352591, E-mail: eldream@e-music.ru*

Alexander V. Chernodarov *- Zhukovsky Air Force Engineering Academy, candidate of science, docent, P.O.Box: Planetnaya St., 125190, Moscow, Russia, E-mail: chernod@mail.ru*

ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ОБЪЕКТОВ ИМИТАЦИОННОЙ МОДЕЛИ, ОСНОВАННОЕ НА ЗНАНИЯХ

Александр Миков, Елена Замятина, Константин Осмехин

***Abstract**: В докладе представлена подсистема балансировки системы имитации Triad.Net,. приводится архитектура системы балансировки, обсуждаются особенности реализации статической, динамической автоматической балансировки и предлагается применять динамическую управляемую балансировку для равномерного распределения объектов имитационной модели по узлам вычислительной системы. В докладе рассматриваются также лингвистические средства языка Triad для описания алгоритмов балансировки.*

***Keywords**: Распределённые вычисления, распределённое имитационное моделирование, статическая балансировка, динамическая балансировка, экспертные системы.*

***ACM Classification Keywords**: I.6 Simulation and Modeling I.6.8 Types of Simulation - Distributed : I.2 Artificial Intelligence I.2.5 Programming Languages and Software - Expert system tools and techniques*

Введение

В настоящее время актуальным является использование высокопроизводительных вычислительных средств для решения всё более усложняющихся задач. Такая необходимость возникает и в имитационном моделировании. С одной стороны это вызвано увеличением объёмов вычислений и желанием оптимизировать время проведения имитационных экспериментов[1,2,3]. Объекты имитационной модели распределяются по вычислительным узлам (кластера, локальной или глобальной сети) и выполняются параллельно, обычно взаимодействуя друг с другом. С другой стороны, использование распределённых имитационных вычислений диктуется необходимостью объединить уже готовые имитационные модели[1] или необходимостью коллективной работы удалённых пользователей для проведения совместных имитационных экспериментов.

Гетерогенность вычислительной среды и гетерогенность имитационной модели являются причиной того, что нагрузка вычислительных узлов и линий связи в ходе имитационного эксперимента может стать несбалансированной. Гетерогенность вычислительной среды объясняется разной производительностью вычислительных узлов и разной пропускной способностью линий связи, а гетерогенность имитационной модели тем, что некоторые объекты могут длительное время находиться в ожидании, в то время, как другие запускают одно событие за другим, всё время выполняя вычисления. Кроме того, некоторые объекты могут интенсивно обмениваться информацией, а другие – весьма редко.

Дисбаланс нагрузки приводит к тому, что выигрыш от использования распределённых вычислительных ресурсов сводится к нулю.

По этой причине возникает необходимость в разработке алгоритмов и программных средств, которые стараются сохранить баланс нагрузки на компьютерах. Эти справедливо как для распределённых имитационных экспериментов[2,3], так и для распределённых вычислений в общем случае.

Алгоритм балансировки должен быть оптимален для имитационной модели любой структуры, с любым механизмом продвижения времени. Это является нелёгкой задачей. Усилия большого числа исследователей были направлены на разработку подобных алгоритмов [2,3]. Однако это были алгоритмы, разработанные для определённого класса задач, или алгоритмы, которые в значительной степени затрагивали код программ[2]. Из вышесказанного следует, что необходим новый подход к балансировке загрузки. Авторы [5]предлагают использовать динамическую балансировку, основанную на знаниях. В докладе обсуждается постановка задачи балансировки при выполнении имитационного эксперимента,

краткий обзор алгоритмов балансировки (в том числе и алгоритмов, применяемых в распределённых системах имитации), архитектура программных средств балансировки. Далее авторы представляют архитектуру подсистемы балансировки в разрабатываемой распределённой системе имитации Triad.Net и предлагают языковые средства Triad для описания алгоритмов балансировки.

Балансировка загрузки вычислительных узлов во время имитационного эксперимента

Чаще всего в параллельном дискретно-событийном имитационном моделировании (PDES-Parallel Discreet Event Simulation) компоненты моделируемой системы представляют собой логические процессы ($LP_i, i=1 \div n$), которые могут функционировать параллельно. Логические процессы распределяются между вычислительными узлами (кластера или сети), взаимодействие процессов осуществляется путём посылки сообщений от одного процесса другому. Во время выполнения имитационного эксперимента возникает конфликт между сбалансированным распределением объектов (логических процессов) по вычислительным узлам и низкой скоростью обменов сообщениями между процессами по линиям связи (линии связи перегружены или имеют низкую пропускную способность). Если логические процессы распределены между процессорами таким образом, что издержки на коммуникацию между ними сведены к нулю, то некоторые процессоры (компьютеры) могут простаивать, в то время как остальные будут перегружены. В другом случае, «хорошо сбалансированная» система потребует больших затрат на коммуникацию. Следовательно, стратегия балансировки должна быть таковой, чтобы процессоры (компьютеры) были загружены достаточно равномерно, но и коммуникационная среда не должна быть перегружена.

Следует различать *статическую* и *динамическую* балансировки. Статическая балансировка выполняется до начала имитационного прогона. При распределении логических процессов по процессорам используется опыт предыдущих имитационных прогонов (именно так происходит предварительное размещение процессов по компьютерам в SPEEDES[2]). В Triad.Net при размещении объектов имитационной модели по узлам ВС используют ещё и структурные особенности имитационной модели: объекты и подобъекты имитационной модели (а модель является иерархической) стараются расположить на одном вычислительном узле. Кроме того, выявляют клики в графе имитационной модели и они также отображаются на один вычислительный узел для того, чтобы сократить время на пересылку сообщений между узлами ВС.

Однако предварительное размещение логических процессов по процессорам (компьютерам) не всегда эффективно.

Это объясняется тем, что:

- Модель во время имитационного прогона может измениться (планирование новых событий, появление новых процессов, завершение работы процессов).
- Может измениться и вычислительная среда, в которой происходит моделирование (выход из строя компьютера или процессора).
- Компьютер (или процессор), на котором выполняется имитационная модель, занят ещё и другими вычислениями, вследствие этого доля работ не связанных с имитационной моделью, может возрасти.

Так или иначе, очень часто выигрыша от предварительного распределения логических процессов по компьютерам с целью выполнения параллельной обработки часто не наблюдают.

Динамическая балансировка предусматривает перераспределение вычислительной нагрузки на узлы *во время имитационного прогона*. В динамической балансировке можно выделить следующие этапы: *оценку* загрузки вычислительных узлов; *инициацию балансировки* загрузки, *принятие решений* о балансировке; *перемещение* объектов с одного вычислительного узла на другой.

Решение о переносе логических процессов (объектов имитационной модели) с одного узла на другой принимается на основании собранных во время имитационного прогона данных (этап оценки загрузки). Эта информация хранится в базе данных и, как правило, содержит данные двух типов:

- об имитационной модели (частота обменов между логическими процессами, количество объектов модели, располагающихся на одном вычислительном узле, продолжительность выполнения того или иного логического процесса и т.д.);
- о состоянии вычислительной системы, на которой выполняется вычислительный эксперимент (данные о загрузке вычислительного узла, о его простоях, о фоновой загрузке, о загрузке линий связи, о промежутке времени, потраченном на посылку сообщения и т.д.).

Кроме того, важно владеть информацией о том, каким образом выполняется обмен информацией между процессами, т.е. топологию обменов (или модель коммуникаций между объектами).

Оценку загрузки процессоров можно оценить аналитически (на основе знаний о поведении модели) или на основании измерений. Большинство современных машин снабжено счетчиками времени (с точностью до микросекунд), которые могут быть использованы для измерения времени выполнения каждой задачи. Сбор данных выполняется специальным программным обеспечением.

Далее следует определить, существует ли дисбаланс и принять решение о проведении перераспределения нагрузки (очень часто проводимая балансировка может привести к отсутствию выигрыша от перераспределения нагрузки).

Дисбаланс загрузки можно определить:

- *Синхронно.* Все вычислительные узлы прерывают работу в определенные моменты синхронизации и определяют дисбаланс загрузки путем сравнения загрузки отдельного процессора с общей средней загрузкой.
- *Асинхронно.* Каждый вычислительный узел хранит историю своей загрузки. В этом случае момент синхронизации для определения степени дисбаланса отсутствует. Вычислением объема дисбаланса занимается фоновый процесс, работающий параллельно с приложением.

Принятие решения о балансировке может быть выполнено:

- *Централизованно*–специальный компьютер собирает глобальную информацию о состоянии всей вычислительной системы и принимает решение о перемещении задач для каждого из вычислительных узлов.
- *Распределенно*–каждый вычислительный узел выполняет собственный алгоритм балансировки, перемещение объектов выполняется только с загруженного вычислительного узла на соседние

Последним этапом является этап перемещения объектов, при этом следует обеспечивать целостность состояния объекта.

Итак, можно сделать вывод о большом количестве алгоритмов и стратегий, используемых при выполнении этапов балансировки нагрузки. Тем не менее, общими являются обозначенные выше этапы балансировки и архитектура программного обеспечения, отвечающего за балансировку нагрузки вычислительных узлов. Перечислим компоненты этого программного обеспечения: программные средства, обеспечивающие оценку состояния распределённой имитационной модели и вычислительной среды (подсистема анализа); управляющая программа, принимающая решение о моменте проведения балансировки, и о том, какие логические процессы следует переместить с одного процессора на другой; программные средства, реализующие перемещение объекта с процессора на процессор, подсистема визуализации, отображающая распределение компонентов имитационного моделирования по вычислительным узлам, коммуникационную среду, изменение состояния имитационной модели и вычислительной среды; базу данных, которая хранит информацию о компонентах имитационной модели и о вычислительной среде.

Теперь рассмотрим представление имитационной модели в Triad.Net и принципы построения подсистемы балансировки в Triad.Net.

Имитационная модель в Triad

Имитационная модель в Triad [4] представляет собой совокупность объектов, которые действуют по определённым сценариям и обмениваются информацией друг с другом и может быть представлена тройкой: μ = {Str, Rout, Mes}.{Str), (Rout), (Mes) – это слой структур , рутин и сообщений соответственно.

Слой структур предназначен для описания моделируемых объектов и связей между ними, слой рутин представляет собой набор алгоритмов поведения моделируемых объектов, а слой сообщений даёт возможность описывать сообщения сложной структуры. Моделируемые объекты часто имеют иерархическую структуру. Имитационная модель также является иерархической. Каждый из уровней можно описать как граф с полюсами P = {U,V,W}, где V – множество вершин графа, каждая вершина представляет собой моделируемый объект, который находится на конкретном уровне иерархии. W – набор дуг, связывающих вершины графа (моделируемые объекты). U – набор внешних полюсов. Внутренние полюса используют для передачи сообщений на одном уровне иерархии. Их разделяют на входные In(V) и выходные Out(V). Набор внешних полюсов служит для передачи информации объектам, находящимся на различных (смежных) уровнях иерархии.

Рутина представлена множествами событий (E), состояний Q, моментов времени. Каждое состояние определяется набором значений локальных переменных (множество Var) каждой конкретной рутины. Система имитации Triad.Net является параллельной дискретно-событийной системой имитации (PDES – Parallel Discrete Event Simulation), в которой события планируют друг друга. Множество событий может быть представлено в виде графа запланированных событий, каждая вершина которого $e_i \in E$.

Для сбора статистических данных о ходе моделирования, для анализа и представления результатов имитационного эксперимента в Triad используют специальные средства – информационные процедуры и условия моделирования. Информационные процедуры и условия моделирования реализуют алгоритм исследования. Алгоритм исследования отделён от модели. Исследователь имеет возможность изменить алгоритм исследования в ходе моделирования, при этом модель остаётся неизменной, нет необходимости вносить в неё какие-либо изменения, чтобы указать алгоритму исследования те элементы модели, за изменением которых надо вести наблюдение.

Необходимо отметить особенность имитационных моделей в Triad: модель не является статической. В Triad определены операции над моделями в каждом из трёх слоёв[4]. Это операции в слое структуре: добавление и удаление вершины, добавление и удаление полюсов, добавление и удаление дуг, рёбер, объединение графов (модель представлена в виде графа), пересечение графов и т.д. В слое рутин – это добавление и удаление событий из графа событий. В слое сообщений – добавление и удаление типов и т.д. Т.е., структура имитационной модели, логика поведения могут быть изменены динамически во время имитационного прогона, а это ещё раз подтверждает необходимость применения алгоритма динамической балансировки.

Подсистема балансировки нагрузки в Triad.Net

Подсистема балансировки вычислительной нагрузки предназначена для оптимального размещения объектов имитационной модели по узлам ВС с целью повышения производительности системы имитации.

Задача балансировки ставится как задача отображения неизоморфных связных графов, B: $TM \rightarrow NG$, где TM – множество графов моделей, NG – множество графов – конфигураций компьютерной сети. Граф $G \in NG$, G = {C, Ed}, определяется множеством вычислительных узлов C и множеством ребер Ed, обозначающих линии связи. Можно рассматривать NG как суперграф, содержащий все возможные (допустимые) графы G в качестве подграфов. Граф $M \in TM$, M = {U, V, W} , задает имитационную модель.

Рассматриваются три разновидности задачи балансировки[5]: статическая Bs, динамическая (автоматическая) Ba и динамическая (управляемая) Bc.

В алгоритме статической балансировке Bs наилучшим результатом считается отыскание подграфа $G \subset NG$, изоморфного графу – модели M. Однако такой подграф существует далеко не всегда, поэтому предлагается метод отыскания в некотором смысле «близкого» подграфа.

В алгоритме автоматической динамической балансировки Ba графы G и M рассматриваются как нагруженные. Вершинам первого графа приписывается параметр – производительность, а его ребрам – скорость передачи данных. Во втором графе вершины характеризуются временной сложностью вычислений, а ребра – интенсивностью потоков сообщений (выходных событий).

Веса вершин и ребер графа NG (и, значит, любых его подграфов) считаются известными. Соответствующие параметры графа M должны определяться во время имитационного прогона. В

соответствии с некоторым алгоритмом происходит определение «узких» мест ВС и имитационной модели и выполняется перенос объектов на менее загруженные узлы, не прерывая процесса моделирования. Алгоритм автоматической балансировки можно описать на языке Triad, пример приведён ниже.

На основании собранной статистики, используя генетические алгоритмы, для последующих имитационных прогонов конкретной модели можно найти лучшее распределение объектов по вершинам графа G. Автоматическая динамическая балансировка использует «прошлое» процесса имитационного моделирования для планирования его будущего выполнения. Однако выполнение конкретной модели может не соответствовать этому предсказанию. Действительно, опыт применения множества алгоритмов автоматической динамической балансировки не даёт заметного выигрыша в производительности. Именно автор модели может знать, как модель может вести себя в том или ином случае: например, автор модели знает, что поток заявок на конкретном устройстве будет наиболее интенсивным через 600 единиц модельного времени. Другой пример: частота обменов между двумя конкретными узлами будет наиболее интенсивной примерно через 300 единиц модельного времени. Таким образом, становится ясным, что для эффективной балансировки необходимы специальные программные и языковые средства, которые позволили бы управлять процессом балансировки.

В Triad.Net предложен алгоритм динамической балансировки, основанной на знаниях.

Управляемая динамическая балансировка использует экспертный компонент и нестандартные информационные процедуры. В экспертном компоненте сосредоточены правила оптимизации, которые формулирует автор для данной модели (или класса моделей). Нестандартные информационные процедуры также разрабатываются автором модели и предназначены для вычисления моментов (или условий) применения правил.

Архитектура подсистемы управляемой балансировки

Итак, подсистема управляемой балансировки включает следующие компоненты:

- *Экспертный компонент*, содержащий базу знаний с правилами для оптимального размещения объектов имитационной модели по вычислительным узлам, редактор правил для модификации этих правил, механизм вывода, компонент объяснения и т.д.
- *Подсистему анализа имитационной модели и вычислительной системы*, на которой выполняется имитационный эксперимент. Подсистема анализа представлена информационными процедурами для сбора информации о поведении объектов имитационной модели (они определяют частоту обменов между объектами, частоту выполнения тех или иных событий и т.д.) и системных информационные процедур, вычисляющих загрузку узлов ВС, пропускную способность линий связи и т.д.
- *Визуализатор модели и вычислительной системы.* Визуализатор отображает статистические данные в виде графиков и диаграмм, причём пользователь может самостоятельно выбрать представление информации, используя те или иные информационные процедуры. Кроме того, визуализатор отображает распределение объектов имитационной модели на вычислительные узлы ВС.
- *Миграционную подсистему,* выполняющую перенос объектов имитационной модели с одного узла ВС на другие.

Как уже упоминалось ранее специальные объекты Triad.Net – информационные процедуры - ведут наблюдение за элементами модели, а именно, отслеживают изменение локальных переменных рутин, фиксируют наступление того или иного события и приход или отсылку сообщения с полюсов вершины. Информационные процедуры могут анализировать и сопоставлять данные от разных объектов модели во время имитационного прогона. Стандартные информационные процедуры входят в подсистему анализа Triad.Net. Кроме того, пользователь имеет возможность написать на языке Triad нестандартную информационную процедуру.

Подсистема балансировки использует информационные процедуры для визуализации хода моделирования, для наблюдения за характеристиками модели. Как только характеристики модели примут критическое значение, они передают сигнал экспертной системе, которая на основании правил выполняет операции над графом G –графом структуры модели, отображённой на граф M – граф ВС.

Правила преобразований могут быть такими:

- *если на i-ом вычислительном узле находится больше 2-х активных объектов, а j-й узел свободен, то перенести один из объектов на j-й свободный узел;*
- *если два объекта V_i и Vj в сети обмениваются данными и в сети есть линия связи Edk с большей пропускной способностью, то отобразить дугу wij на эту линию связи;*
- *если два объекта (V_i и V_j) в сети интенсивно обмениваются данными и в сети есть незанятый узел V_i, то переместить их на этот узел и т.д.*

Таким образом, правила сводятся к выполнению операций над графом G. Правила представляют собой продукции вида «if …then…else…».и могут быть описаны на языке Triad. На языке Triad может быть описан и алгоритм автоматической (неуправляемой) балансировки.

Языковые конструкции языка Triad для описания алгоритма балансировки

Приведём фрагмент описания алгоритма автоматической балансировки, выполненный на языке Triad. Этот алгоритм был использован в распределённой системе имитации SPEEDES[3]. Следуя этому алгоритму, при возникновении дисбаланса нагрузки выбирают наиболее загруженный узел. Далее возникает необходимость в выборе одного из объектов имитационной модели, находящихся на этом узле и переносе его на другой, менее загруженный узел. Выбор объекта выполняется случайным образом.

Описание имитационной модели начинается с ключевого слова *model* языка Triad и завершается ключевым словом *endmod*. Описание модели включает описания слоя структур имитационной модели (*structure …endstr*), описание слоя рутин (*routine … endrout*(поведение объектов модели)) и слоя сообщений *message…endmsg* (сообщения сложной структуры). Пусть структура имитационной модели G представляет собой граф с вершинами A,B,C,D,E,F (полный граф с вершинами A,B,C,D и двумя присоединёнными к вершине D вершинами E и F. Вершины также соединены ребом лруг с другом).

model Mod1;

var graph G; *structure* S *def*…G1:=compl(A,B,C,D)+node(E)+Node(F)+edge(E<>F)+edge(D<>E)+ edge(D<>F) …endstr

…G:=S; (* описали структуру имитационной модели, compl – графовая константа «полный граф»*)

Для описания ВС, с помощью которой выполняется имитационный эксперимент, также следует использовать слой структур языка Triad.

….*Var graph* M; (* описание ВС, которая представляет собой сеть топологии звезда, в сети 3 компьютера и в центре – компьютер О *) *structure* S1 def … G2:=*star*(3)(O,P,R,S) …*endstr*; M :=S1: …(*структура ВС*).

Для описания алгоритма статической балансировки используем процедуру, в которой пользователь может определить предварительное размещение объектов имитационной модели по вычислительным узлам. Это процедура должна быть выполнена до начала имитационного эксперимента, поэтому ссылку на процедуру следует разместить в части *conditions of simulation (условия моделирования)*. Эта часть программного кода реализуется в первую очередь. Процедура статической балансировки может выглядеть следующим образом:

procedure Static_Load_Balancing (*in ref node* G1, *ref node* V,W,X,Y,Z) *def* X:=G1; Y:=V; Z:=W;*endproc*

Описание автоматической балансировки выполняется процедурой Automated_Load_Balancing. Алгоритм, описываемый этой процедурой, выбирает случайным образом один из объектов имитационной модели, размещённый на вычислительном узле, который указывается в качестве входного параметра процедуры, и выполняет перенос этого объекта на другой вычислительный узел. Этот узел также передаётся в качестве входного параметра. Узлы определяются системными информационными процедурами как наиболее и наименее загруженные. Пусть это будут узлы M.P и M.R.

procedure Automated_Load_Balancing (in ref node X,Y) def

set U *of node* (G.A,G.B,G.C,G.D); *ref node* Q; Q:= *random*(U); Migrate(Q) *from* (X) *to* (Y)

endproc

Ссылку на процедуру Automated_Load_Balancing также размещают в условиях моделирования (в основной части после вызовов информационных процедур).

Conditions of simulation Running …

Initial … Static_Load_Balancing(compl(G.A,G.B,G.C,G.D,G),G.E,G.F,M.P,M.R,M.S) …*endi*

….(*вызов информационных процедур для сбора статистики*)…

(*вызов процедуры автоматической балансировки*)

Automated_Load_Balancing (M.P,M.R); *processing* ….*endproc*;

Endcond

Заключение

Итак, в статье описана архитектура подсистемы балансировки и языковые конструкции для описания алгоритмов балансировки. Языковые конструкции дают возможность пользователям распределённой системы имитации Triad.Net написать свой алгоритм балансировки, который наиболее адекватен поведению имитационной модели, над которой они проводят имитационный эксперимент. Подсистема балансировки наряду со статической и автоматической балансировками включает также программные средства для проведения управляемой балансировки, основанной на правилах. Правила задаёт пользователь, знающий особенности поведения конкретной имитационной модели

Благодарности

Работа выполнена при поддержке гранта РФФИ 07-07-00412-а

Библиографический список

1. [Fujimoto, 2003] Fujimoto R.M. Distributed Simulation Systems. In Proceedings of the 2003 Winter Simulation Conference S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds. pp. 124-134
2. [Wilson, 1998] Wilson L.F. and Wei Shen. Experiments In Load Migration And Dynamic Load Balancing In Speedes. Proceedings of the 1998 Winter Simulation Conference.D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds, pp.590-596
3. [Zheng, 2005] Zheng G. Achieving High Performance on Extremely Large Parallel Machines: Performance Prediction and Load Balancing; in Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005,165p. Доступно на сайте: http://charm.cs.uiuc.edu/
4. [Mikov, 1995] Mikov A.I. Simulation and Design of Hardware and Software with Triad// Proc.2nd Intl.Conf. on Electronic Hardware Description Languages, Las Vegas, USA, 1995. pp. 15-20.
5. [Миков, 2005] Миков А.И.., Замятина Е.Б., Осмехин К.А. Метод динамической балансировки процессов имитационного моделирования. В кн. «Материалы Всероссийской научно-технической конференции «Методы и средства обработки информации МСО-2005». М.: Изд-во МГУ, 2005, стр.472-478.

Сведения об авторах

Александр Миков *– АНО «Институт компьютинга», директор; Россия, г. Краснодар, ул. Аксайская, 40/1-28; e-mail:*

Елена Замятина *– Пермский государственный университет, доцент кафедры математического обеспечения вычислительных систем, Россия, г. Пермь, 614017, ул. Тургенева, д. 33, к. 40; e-mail:*

Константин Осмехин *– Пермский государственный университет, аспирант кафедры математического обеспечения вычислительных систем, Россия, г. Пермь, 614017, ул. Гашкова, 28-12; e-mail: kosmehin@lukoilperm.ru ; e_zamyatina@mail.ru ; alexander_mikov@mail.ru*

CULTURE ASPECTS OF INFORACTION

Krassimir Markov, Stoyan Poryazov, Krassimira Ivanova, Ilia Mitov, Vera Markova

***Abstract**: The adequate attitude to the information models and information objects in the culture context is one of the main problems to be investigated on the threshold of information society. The goal of this paper is to outline some problems connected with the main styles of perceiving of the mental and artificially generated information models stored in the information objects and used in the processes of the Information Interaction or simply – in the **Inforaction**. The culture influence on inforaction is discussed.*

***Keywords**: General Information Theory, Inforaction, Information Models, Artificial Information Models*

***ACM Keywords**: A.1 Introductory and Survey*

Introduction

The world common information bases make possible to exchange information of any kind. Some information could not be proved easy, some is assumed as "clear". In addition, now we have a new phenomenon – artificially created information objects which need to be treated in eligible way.

What is the proper approach to the perceiving of the ocean of information we exchange during the information interaction?

The adequate attitude to the information objects in the culture context is one of the main problems to be investigated on the threshold of information society. The information interaction is not isolated process. All culture phenomena influence to the styles of perceiving the information objects. The interrelations between two main opposite – scientific and non-scientific – styles of perceiving the information models need to be discussed.

The investigation in this paper is provided from point of view of the Theory of Inforaction (a part of the General Information Theory (GIT) [Markov et al, 2007]). The goal of this paper is to outline the main styles of perceiving of the mental and artificially generated information models stored in the information objects used in the processes of the Information Interaction or simply – the ***Inforaction***.

Our further explanation needs of remembering some basics from the General Information Theory (GIT) [Markov et al, 2007].

The information models

The concept "**model**" has been used for denotation of the very large class of phenomena: mechanical, theoretical, linguistic, etc. constructions. Marx Wartofsky gave a good definition of the model relation and made clear the main characteristics of the model [Wartofsky, 1979]. This definition is as follow:

The model relation is triple M:

$$M: (S, x, y)$$

where "S" is subject for whom "x" represents "y". In other words only in this relation and only for the subject "S" the entity "x" is a model of the entity "y".

As we point in [Markov et al, 2007], the interaction between two entities is a specific theirs relationship. If there exist information witness (**W**) of the interaction between two entities as well as of the existence of the information about the first entity (A) in the second entity (B), **W** became as subject for whom the information in the second entity (B) represents the first one (A) (the information of A in B represents A). In other words, there exists relation

$$M: (\mathbf{W}_{BA}, I_{BA}, A),$$

where "A" and "B" are entities, and the $\mathbf{W}_{BA}$ is the information witness, which proofs that the assertion "$I_{BA} \subset B$ is information in B for A" is true. In the relation ($\mathbf{W}_{BA}$, I_{BA}, A) the information I_{BA} is a model of A**.**

The entities of the world interact continuously in the time. It is possible, after any interaction, the other one may be realized. In this case, the changes received by any entity, during the first interaction, may be reflected by the new entity. This means the ***secondary (transitive, external) reflection*** exists. The chain of the transitive reflections is not limited.

Let A, B and C are entities. Let A and B interact and after that B interacts with C.

Let there exist the relations:

- M_{BA}: ($\mathbf{W}_{BA}$, I_{BA}, A), where $\mathbf{W}_{BA}$ is the information witness, which proofs that the assertion "$I_{BA} \subset B$ is information in B for A" is true; i.e. I_{BA} is information of A in B.
- M_{CB}: ($\mathbf{W}_{CB}$, I_{CB}, B), where $\mathbf{W}_{CB}$ is the information witness, which proofs that the assertion "$I_{CB} \subset C$ is information in C for B" is true; i.e. I_{CB} is information of B in C.
- $M_{C(B)A}$: ($\mathbf{W}_{C(B)A}$, $I_{C(B)A}$, A), where $\mathbf{W}_{C(B)A}$ is the information witness, which proofs that the assertion "$I_{C(B)A} \subset C$ is information in C for information in B for A" is true; i.e. $I_{C(B)A}$ is transitive information of A in C.

In such case, from point of view of the $\mathbf{W}_{C(B)A}$ the information $I_{C(B)A}$ is a model of A. In other hand, because of transitive reflection, $I_{C(B)A}$ is created as reflection of the sign I_{BA} but not directly of A. This means that $I_{C(B)A}$ is a model of the information in B for A, i.e. $I_{C(B)A}$ is an ***information model*** in C for A [Markov et al, 2001].

The collecting of information models for given entity in one resulting entity may exist as a result of the process of interaction between entities. Such process is in the base of the ***Information modeling***.

The possibility of self-reflection may cause the generating the new information models in the memory without any external influence.

Information Objects and Processes

The entity, which has possibility for:

- *(primary) activity* for external interaction;
- *information reflection and information memory*, i.e. possibility for collecting the information;
- *information self-reflection*, i.e. possibility for generating "secondary information";
- *information expectation* i.e. the (secondary) information activity for internal or external contact;
- *information modeling and resolving the information expectation*

is called Information Subject or **Infos** [Markov et al, 2007].

An entity, in which one or more information models are reflected, is called ***"information object".***

The information objects are only tools for the information exchange in the space and time, i.e. for the realizing the information interaction.

The information objects can have different properties depending on:

- the kind of influence over the entities - by ordering in space and time, by modifying, etc.,
- the way of influence over the entities - by direct or by indirect influence of the Infos on the object,
- the way of development in time - static or dynamic,

etc.

It is clear, that the Infoses are information objects.

The information is kind of indirect reflection. The only way one to operate with information is to operate with the entity it contains. An action on the entity may cause any internal changes in it and this way may change the information already reflected. The influence over the information object, regarding the contained information, is called "***information operation***".

The information operations may be of two main types:

- the Infos internal operations with the sub-entities that contain information,
- external operations with the information objects that contain information.

The internal operations with the sub-entities closely depend of the Infos possibilities for self-reflection and internal interaction of its sub-entities. The self-reflection (self-change) of the Infos leads to the creating of new relationships (and corresponding entities) in it. These are subjectively defined relationships, or shortly – ***subjective relationships***. When they are reflected in the memory of the Infos they initiate ***subjective*** information model. These subjective information models may have not real relationships and real entities that correspond to them. The possibility for creating the relationships of similarity is a basis for realizing such very high level operations as "comparing elements or substructures of the information models", "searching given substructure or element pattern in the part or in the whole structure of the information model", etc.

The external operations with information objects may be differed in two main subtypes – basic and service operations.

There are two "***basic information operations***" which are called I-operations:

- I-reflection (reflecting the information object by the Infos, i.e. the origination of a relevant information model in the memory of the Infos).
- I-realization (creating the information object by the Infos);

In the process of its activity, the Infos reflects (perceives) information from the environment by proper sub-entities (sensitive to video, acoustic, tactile, etc. influences) called "***receptors***". Consequently, the Infos may receive some information models. This subjective reflection is called "***I-reflection***".

When necessary, the Infos can realize in its environment some of the information models, which are in his memory, using some sub-entities called "***effectors***". Consequently, new or modified already existing entities reflect information, relevant to these information models. This subjective realization is called ***"I-realization"***.

There are several operations, which can be realized with the information objects: transfer in space and time, destroying, copying, composition, decomposition, etc. Because of the activity of the Infoses, these operations are different from other events in reality. In this case, the Infos determined operations with information objects are called "***service information operations***".

Let t_1, t_2,..,t_n are information operations. The consequence of information operations P, created using the composition, i.e.

$$P = t_1 \circ t_2 \circ \ldots \circ t_n$$

is called "***information process***". In particularly an information process can include only one operation.

The Information Societies

If an information model from an Infos is reflected in another entity, there exist possibility, during the "a posterior" interactions of the given entity with another Infos, to transfer this reflection in it. This way an information model may be transferred from one Infos to another.

Let S_1 and S_2 are Infoses and O is an arbitrary entity. The composition of two contacts

$$S_1 \xrightarrow{\Theta_{S_1O}} O \xrightarrow{\Theta_{OS_2}} S_2 \qquad (1)$$

is called "***information contact***" between Infos S_1 and Infos S_2 iff during the contacts any information model from S_1 is reflected in the Infos S_2 true the entity O. The Infos S_1 is called "***information donor***", the Infos S_2 is called "***information recipient***", and the entity O is called "***information object***".

For the realization of one information contact at least one information object is necessary. This way the elementary communicative action will be provided. In general, every information process "k", having as a start domain the set S_d (of information models) and as a final domain the set S_r (again of information models), which may be coincidental, we call "information contact": k: $S_d \rightarrow S_r$. S_d is called "Infos-donor" and S_r - "Infos-recipient".

The set "R" of all information contacts between two Infoses S_a and S_b : R= $\{k_i \mid i=1,2..; k_i:S_a \rightarrow S_b\}$ is called **"*information interaction*"** or simply "***inforaction***".

When S_a and S_b are coincident, we call it Information interaction with itself (in space and time). The set "B" of all information objects, used in the information interaction between given Infoses is called "***information base***".

A set of Infoses is called **"Society",** iff there exists agreement for information interaction between them, by means of which they could communicate. An important element of this agreement is the availability of a common information base. In other words, every group of information subjects, people in particular, is a society if any agreement for information interaction between them exists.

This definition is in accordance with usual understanding of the concept "society". The sociologist Richard Jenkins remarked that the term addresses a number of important existential issues facing people [Jenkins, 2002]:

1. How humans think and exchange information – the sensory world makes up only a fraction of human experience. In order to understand the world, we have to conceive of human interaction in the abstract (i.e., society).
2. Many phenomena cannot be reduced to individual behavior – to explain certain conditions, a view of something "greater than the sum of its parts" is needed.
3. Collectives often endure beyond the lifespan of individual members.
4. The human condition has always meant going beyond the evidence of our senses; every aspect of our lives is tied to the collective.

We shouldn't picture the information base like a number of drives with a certain data recorded, although it's the way it's been since the beginning – it was recorded on clay plates, papyrus, paper. The ability for digital storage of the data lays the beginnings of the genesis of the "**Information Societies**".

It's obvious that, there are many societies with correspondent information bases, and a person could belong to more than one society. Thus we could talk about "information societies" which exist in a certain way with or without a particular correlation between them. And it's not very likely for the humanity to reach such state of integrity so we could use this term in singular when speaking about the population of the whole planet. Nevertheless the concept "global information society" is very popular. This is a general concept which means the hypotetic digitally based integrated humanity.

The Culture Environment

The concept "Culture" means "every aspect of life: know-how, technical knowledge, customs of food and dress, religion, mentality, values, language, symbols, socio-political and economic behavior, indigenous methods of taking decisions and exercising power, methods of production and economic relations, and so on" [Verhelst, 1990].

The culture permeates and influences every aspect of life, but it is not static however, rather it is a process in a constant state of flux and adaptation to new contexts, demands, and needs. Culture is not a deterministic force but rather a subtle and often subliminal pattern of thinking that describes the "organization of values, norms, and symbols which guide the choices made by actors, limit the types of interaction and may occur between individuals" [Parsons et al, 1990].

Culture is "learned, and shared. In addition, culture is adaptive. Human beings cope with their natural and social environment by means of their traditional knowledge". In other words, as something inherited, 'traditional' cultural knowledge developed within a particular spatial and temporal "context" or "environment". But as a dynamic process culture continues to change as people cope with new challenges and adapt to changing conditions. Underlying values and expectations are arbitrary conceptions "of what is desirable in human experience, ... (and) these concepts of what is desirable combine cognitive and affective meanings ... they provide security and contribute to a sense of personal and social identity. For this reason, individuals in every society cling tenaciously to the values they have acquired and feel threatened when confronted with others who live according to different conceptions of what is desirable". Thus culture is like a "security blanket" which "has great meaning to its owner" [Spradley, McCurdy, 1987].

"Culture is at once socially constituted (it is a product of present and past activity) and socially constitutive (it is part of the meaningful context in which activity takes place)" [Roseberry, 1989].

A diversity of specific culture concepts was grouped into different categories and shown in table 1 as follows.

Table 1: Different definitions of culture ([Cultural Capital, 2003])

Definitions	
Topical	: Culture consists of everything on a list of topics, or categories, such as social organization, religion, or economy
Historical	: Culture is social heritage, or tradition, that is passed on to future generations
Behavioral	: Culture is shared, learned human behavior, a way of life
Normative	: Culture is ideals, values, or rules for living
Functional	: Culture is the way humans solve problems of adapting to the environment or living together
Mental	: Culture is a complex of ideas, or learned habits, that inhibit impulses and distinguish people from animals
Structural	: Culture consists of patterned and interrelated ideas, symbols, or behaviors
Symbolic	: Culture is based on arbitrarily assigned meanings that are shared by a society

The anthropologist Leslie White (1900-1975) suggested that for analytical purposes, a culture could be viewed as a three-part structure composed of subsystems that he termed ideological, technological, and sociological. In a similar classification, the biologist Julian Huxley (1887-1975) identified three components of culture: mentifacts, artifacts, and sociofacts. Together, according to these interpretations, the subsystems comprise the system of culture as a whole. But they are integrated; each reacts on the others and is affected by them in turn [Fellmann et al, 2007].

– ***Mentifacts***: The ideological subsystem consists of ideas, beliefs, and knowledge of a culture and of the ways in which these things are expressed in speech or other forms of communication. Mythologies and theologies, legend, literature, philosophy, and folk wisdom make up this category. Passed on from generation to generation, these abstract belief systems, or mentifacts, tell us what we ought to believe, what we should value, and how we ought to act. Beliefs form the basis of the socialization process. Often we know (or think we know) what the beliefs of a group are from their oral or written statements. Sometimes, however, we must depend on the actions or objectives of a group to tell us what its true ideas and values are. "Actions speak louder than words" and "Do as I say not as I do" are commonplace recognitions of the fact that actions, values, and words do not always coincide.

– ***Artifacts***: The technological subsystem is composed of the material objects, together with the techniques of their use, by means of which people are able to live. Such objects are the tools and other instruments that enable us to feed, clothe, house, defend, transport, and amuse ourselves. We must have food, we must be

protected from the elements, and we must be able to defend ourselves. Huxley termed the material objects we use to fill these basic needs artifacts.

– ***Sociofacts***: The sociological subsystem of a culture is the sum of the expected and accepted patterns of interpersonal relations that find their outlet in economic, political, military, religious, kinship: and other associations. These sociofacts define the social organization of a culture. They regulate how the individual functions relative to the group, whether it be family, church, or state. There are no "givens" as far as the patterns of interaction in any of these associations are concerned, except that most cultures possess a variety of formal and informal ways of structuring behavior. Differing patterns of behavior are learned and transmitted from one generation to the next [Fellmann et al, 2007].

It is clear that the sociofacts are variety of information models whith different importance and actuality.

Let remark that the main and most important part of artifacts is formed by the techniques, i.e. the information models of using the material objects. Without this information the material objects are unusable. In addition, without knowledge, without the information models to build material objects they could not became reality. So, we may conclude, that the information models are in the base of artifacts.

The mentifacts, in particulary the esoteric and religious information objects, are important parts of the culture environment. Theirs main characteristic is that they explicitly or implicitly lead to any supernatural phenomena. The exoterics and religions correspond to thousands-years old concepts. Because of this, the discussion is more complicated and needs an example. We may ask ourselves “What is Santa Claus”. From the point of view of our paradigm we could answer: Santa Claus is an information model, which, if followed could achieve very delightful results. That’s why he doesn’t die, as long as there are people who follow the model. It’s not simple but rather a subject with a great variety of personifications – from the jolly old man, who the Coca-Cola Company dressed in red, and the Pepsi Company – in blue, to the vivid character of the Russian Ded Moroz who’s wearing a huge fur-coat, a boyar hat and has a down-to-the-waist beard. Believing in Santa Claus is actually accepting and following of one of the variations of his information model. Every religion is a totality of information models, which are assumed and followed. Many of them are very important for human been and for stability of the social systems.

At the end we need to ask "Where is the difference between the religion and the science, which is also a combination of important information models to be followed?" It is clear – at the first place the difference is the believing in the supernatural phenomena. This leads to the way we create and perceive the information models and the attitude to them. There are two main ways:

– ***The first*** is wonderfully described by the motto of the medieval theologian Anselm of Canterbury, lately canonized as St. Anselm (1033-1109): ***"Credo, ut intelligam!"*** (I believe in order to understand) [St.Anselm]. You have to believe in the information model, so you could understand and follow it. This is the non-scientific approach – every subjective notion can turn into a commonly accepted model or dogma, as long as there’s someone to believe in it and follow it implicitly.

– ***The second*** is described with the phrase ***"Intelligo, ut credam !"*** (I understand in order to believe), used by the German reformer Thomas Muentzer (~1490-1525) [Muentzer]. You have to understand the information model and only after then to trust it if possible. This is the scientific approach – every science builds information models – hypothesizes, which are repeatedly tested before assumed to be true. The scientific approach includes a permanent improvement and revolutions of the existing models [Kuhn, 1966].

The culture aspects of the inforaction

From the point of view of the Theory of Inforaction the cultural environment is the set of all information bases (in the sence given above) which are available in the society. These bases grow permanently because building and exchanging of information models are basic activities for every society. Whether they are perceived with the scientific or non-scientific approach is a question only of the circumstances, executors and users.

In the information contact (1) the Infos S_2 reflects only the information object O, but not the whole process of its genesis. This means that S_2 need to reconstruct in his mind the missing part of the sheme and to make decision what to do with the incoming information model – to accept or not. In this case the important role plays the culture environvent – it may obligate the S_2 to accept O as a dogma or to fill free to make his own decision.

In addition, the information models and objects generated by any artificial systems (Infortrons [Markov et al, 2007]), i.e. so called "***artificial information models and objects***" became one of the main tools for information interaction.

What is the purpose of artificial systems?

– to be a substitute of any of the Information Subjects (Infoses);

– to extent its possibilities to create the information models.

In both cases if an artificial information model is used in the process of information interaction, the perceiver need to decide what attitude he or she needs to assume. The artificially generated information objects may be of any kind and some times it is impossible to make diference between human and artificially generated information object. In addition, in the information processes any information operations may be provided by any artificial systems and the final result may be an information object with mixed genesis.

The receiver's "information immune system" needs to select what from incoming information objects to be verified and what not. Usualy, the scientific oriented subjects do not accept information models which lead to supernatural origins. For every adult person it is clear that Santa Claus does not exist. But the information objects created from other scientists usually are acceped as already verified. The result may be unpredictable. The main peril is the exchange of scientific approach for non-scientific. The scientific information models may be perceived in non-scientific maner. For instance we may point the myth about spinach.

The myth about spinach and its high iron content may have first been propagated by Dr. E. von Wolf in 1870, because a misplaced decimal point in his publication led to an iron-content figure that was ten times bigger than the real. In 1937, German chemists reinvestigated this "miracle vegetable" and corrected the mistake. It was described by T.J. Hamblin in British Medical Journal, December 1981.

The case with the spinach is an example of the unintentional error. But more dangerous are the aforethought actions which may cause damages in the global range. One very significant example in this area is the myth about the fluoride [EWG, 2006].

Fluoride exposure has created controversial health concerns in the United States. For years, doctors and dentists have alleged that fluoride was actually a benefit to health, promoting strong, cavity free teeth. However, studies have suggested that the health risks associated with fluoride exposure may in fact outweigh the benefits. Fluoride exposure has been linked to the development of bone cancer - including osteosarcoma in children - among other serious health complications.

Fluoride is commonly found in or added to numerous consumer products, including tap water, toothpaste, juices, teas, wines, beers, infant formula, sodas, seafood, processed chicken, cigarettes, cereal, anesthetics and Teflon pans. Doctors and dentists have long recommended fluoride exposure for the prevention of tooth caries such as cavities and decay. Accordingly, many municipalities artificially fluoridate their public water supply. Fluoridated water is the greatest source of exposure to fluoride for children.

The federal government first set limits on the amount of fluoride in tap water in 1945. The recommended or optimum level for artificial fluoridation of drinking water was then set at 1 ppm or 1mg/L and remains at that level today. In the 1980s, the United States Environmental Protection Agency revisited the recommendations and raised the maximum contaminant level (or maximum amount of fluoride allowed in water and still considered safe) to 4ppm. Municipalities can independently determine whether to fluoridate their water supplies but cannot exceed the levels set by the federal government. Approximately 60% of all public water is or has been fluoridated.

Recent studies suggest a strong correlation between childhood fluoride exposure and the development of osteosarcoma in young boys. Studies performed by the United States National Toxicology Program and Harvard

University have determined that there is biological and physical evidence relating the development of osteosarcoma cancer to children experiencing fluoride exposure in the bone formative years.

Fluoride is a known mutagen, particularly where it is found in concentrated amounts. In the body, fluoride accumulation occurs primarily in the bones, particularly during the developmental years. There, fluoride artificially stimulates bone cell growth, generally in long bones such as the legs and arms, leading to cancerous growths. Osteosarcoma in children, particularly young boys exposed during the bone growth spurt years of five to ten, has been specifically associated with the effects of fluoride exposure.

Osteosarcoma cancer is characterized by the growth of a cancerous tumor in the bone. The cancer generally occurs in the legs or arms and may cause pain and swelling, broken bones, or a visible lump. Treatment of osteosarcoma, like other cancers, may include a course of chemotherapy and radiation but osteosarcoma is not particularly responsive to radiation. Surgery, and sometimes amputation, is frequently required to treat a patient with osteosarcoma.

Osteosarcoma is not the only serious side effect of fluoride. Bone cancer, bone pain and swelling, and fluorosis have all been associated with excessive fluoride exposure. The effects of fluoride can cause long term and irreversible health effects. Treatment for osteosarcoma and other fluoride induced health problems can be a long and expensive process resulting in physical, emotional and financial stress on the victim and the victim's family.

In spite of this very dangerous data how many persons, asked on the street, will ansver that the fluoride is not useful for the teeth?

The artificially generated information objects are assumed as scientifically generated and as result they are in the same category of "verified in advace" information models. Again, the resul may be unpredictable. Very important examples are the "e-government" information objects. The inaccuracies of the governvent administrators and of the information systems are assumed as it need to be and the result is a great chaos in the business and social activities. Now, in many of the East European countries and especially in Bulgaria this is an every day situation. The culture environment is very susceptible especially in this case and the cultural changes to the worst may be easily recognized in these countries.

Instead of supporting human acivities the e-systems are taken as control and supervising social elements. In the same time, the government officials are assumed as service attendants of the e-systems which are not responsible for theirs activities. The usual saying is not "the law demands …" but the "system demands …".

At the end, the S_2 may be an Infotron. What do we need to take in account in such case? For many years the Infotrons have simple formal reflection (input) and the cultural environment was undefined concept. But the importance of culture environment is obvious. The Infotrons' decisions closely depend on it especially in the cases when they will live in the same information societies with mankind.

Conclusion

The Artificial Intelligence (AI) needs to pay attention to all available information bases, i.e. the culture environment, during modeling the brain activities. We expect the investigations on the boundary of the individual and social intelligence to become in the focus of the AI scientists. The AI models and realizations will take in account the existence of the culture environment. At the first place, the same AI system may give different results in different culture environments.

In abstract theories it is simple to make classifications like – "natural-artificial", but in the real human activities it is important to clear who owns the responsibility. This closely depends on the culture environment. The role and the importance of a particular exoterics and religions in a certain society are determined by the influence of the people ready to doubt the information models, on the others who easily and “blindly” follow the dogmas.

Keeping in mind the limited abilities of the human mind, we can presume that the non-scientific approach would probably dominate. Just a small part of the humanity would be able to build and understand the difficult scientific information models.

The problem is that the artificial information objects may be considered as "dogmas" because the user will perceive it “blindly” without any additional information about its genesis.

That’s why it’s crucial to keep the harmony and dialectical unity of the scientific and non-scientific approaches, following the wisdom of St. Augustine: ***"Intelligo ut credam, credo ut intelligam!"*** [St. Agustine].

Bibliography

[Cultural Capital, 2003] *Youth in Hong Kong. A Statistical Profile 2002*. Report Submitted to Commision on Youth. Social Sciences Research Centre. The University of Hong Kong. March 2003 http://www.info.gov.hk/coy/eng/report/doc/Youth_Statistical/2002/app/Chp6_Cultural_Capital.pdf

[EWG, 2006] Harvard Study: *Strong Link between Fluoridated Water and Bone Cancer in Boys*. Environmental Working Group. April 5, 2006. http://www.ewg.org/issues/fluoride/20060405/index.php (Access 29.05.2007).

[Fellmann et al, 2007] D.Fellmann, A.Getis, J.Getis. *Human Geography: Landscapes of Human Activities*. McGraw-Hill College, 2007. ISBN-13: 9780072827651

[Jenkins, 2002] R.Jenkins. *Foundations of Sociology*. London: Palgrave MacMillan, 2002.ISBN 0-333-96050-5.

[Kuhn, 1966] T.S.Kuhn. *The Structure of the Scientific Revolutions*. Chicago: University of Chicago Press. 1996.

[Markov et al, 2001] K.Markov, P.Mateev, K.Ivanova, I.Mitov, S.Poryazov. *The Information Model*. In: Proceedings of the International Conference KDS-2001 - Sankt-Petersburg, Russia, 2001 - pp. 468-475;
International Journal Information Theories & Applications - Sofia, 2001 - V. 8, No 2 - pp. 59 -69.

[Markov et al, 2007] Kr.Markov, Kr.Ivanova, I.Mitov. *Basic Structure of the General Information Theory*. Int. Journal “Information Theories and Applications”, 2007, Vol.14, No.1, pp.5-19.

[Muentzer] http://www.thomas-muentzer.de/ , http://www.answers.com/topic/thomas-muentzer

[Parsons et al, 1990] T.Parsons, E. Shils. *Values and Social Systems*. In J.Alexander, S.Seidman (ed.), Culture and Society, Contemporary Debates. Cambridge Univ Press, New York 1990. pp.39-40.

[Roseberry, 1989] W.Roseberry, *Anthropologies and Histories*. Rutgers University Press, New Brunswick 1989. p.42.

[Spradley, McCurdy, 1987] P. Spradley, D.W. McCurdy. *Conformity and Conflict: Readings in Cultural Anthropology*. Little Brown and Company, Boston 1987. pp.4-6.

[St.Agustíne] http://www.sant-agostino.it/links/inglese/index.htm, http://www.conoze.com/doc.php?doc=157

[St.Anselm] http://webspace.ship.edu/cgboer/middleages.html , http://maritain.nd.edu/jmc/etext/hop30.htm

[Verhelst, 1990] T.Verhelst. *No Life without Roots*. London: Zed Books 1990 p.17

[Wartofsky, 1979] M.W.Wartofsky. *Models*. Representation and the Scientific Understanding. D.Reidel Publishing Company, Dordrecht: Holland /Boston: USA, London: England/, 1979.

Authors' Information

Krassimir Markov - Institute of Mathematics and Informatics, Bulgarian Academy of Sciences; Institute of Information Theories and Applications FOI ITHEA, markov@foibg.com

Stoyan Poryazov - Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, stoyan@cc.bas.bg

Krassimira Ivanova - Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, kivanova@math.bas.bg

Ilia Mitov - Institute of Information Theories and Applications FOI ITHEA, mitov@foibg.com

Vera Markova - Institute of Information Theories and Applications FOI ITHEA, vera@tu-sofia.bg

MAGIC OF EGREGORS

Vitaliy Lozovskiy

***Abstract**: This paper is continuation of [Lozovskiy, 2003, 2005]. The main issue is: "What esotericism can add to our understanding of human intellect, consciousness, to creation of AI?" The history of esoteric teachings covers several millenniums, while that of AI – less than 50 years. Questioned are even the philosophic fundamentals – idealism vs. materialism, concepts of matter, energy and information. In what follows I try to elucidate notions of intellectual subject, sign, signal, information, belief, mystic theories, egregors. Analysis of such concepts as "magic", "God" forces me to conclusion that they are artifacts of our psychology and socio-psychology. At the same time, real importance of these aspects can be so determinative that efforts to create AI without proper consideration of these issues look impossible.*

***Keywords**: philosophy, noosphere, esoteric, intangible world, beliefs, faith, trust, conviction, soul, God, materialism, idealism, mystic theories, magic, egregors*

***ACM Classification Keywords**: I.2.0 General: Philosophical foundations, H.1.1 Systems and Information Theory: Information theory, I.6.5 Model Development: Modeling methodologies, I.2.11 Distributed Artificial Intelligence - Multiagent systems, I.2.m Miscellaneous, H.1.2 User/Machine Systems*

When one rows, it is not the rowing which moves the boat. Rowing is only a magical ceremony by means of which, one compels a demon to move the boat.

Friedrich Nietzsche

God is real, unless of course, he is declared an integer.

Programmer's humor

Introduction

The history of intensive artificial intelligence (AI) research approaches its half-century mark. Looking at overall explosion-like information technology (IT) progress, one should confess, that the progress in AI, consciousness simulation, intellectual robots drops behind expectations. The first approach to brain mechanisms understanding based on the McCullock-Pitts formal neuron approach failed due too primitive functionality of one threshold logic element. The same bad luck crowned first perceptrons and neural network paradigm. Marvin Minsky proved that primitive neural network assemblies cannot demonstrate serious functionality, while advanced multilayer neuron architectures with rich cross- and feedback connectivity were far beyond available theoretical analysis and algorithmic support. Besides, the hardware of 60-ies was too weak to support implementation of large network models in anticipation of their nontrivial behavior. No quantity – no quality… Dialectics is going strong. Neurophysiologists and psychologists of that era could not discover and explicate constructive principles of human brain mechanisms, so IT engineers were brought to necessity of saving the drowning problem with their own efforts. So, the first success in human intellect simulation was achieved with the help of introspection and heuristic programming. Impressive achievements in this field were due the progress in semiotics, knowledge acquisition and engineering, expert system development, natural language understanding within computers.

The first alert bells rang when massive automated management system creation bumped into the concrete wall of intrinsic difficulties of their development and support. Expert systems approach in AI together with goal directed programming, blackboard and production systems architectures helped to overcome some difficulties, but only

temporarily loosed this Gordian knot. The same Scillas and Haribdas of complexity vs. functionality were in the same place, as before. In order to achieve richer functionality one should increase complexity of his/her solution eventually loosing control over the structure and processes within it… Besides, the problem of knowledge elicitation from application domains and human experts was also under the tremendous press of real practical requirements.

The dramatic progress in computers opened the road towards the second try in neuron network approach. Powerful strategies for training such networks were developed; genetic algorithms and multiagent architectures appeared on the scene. All that helped AI and, in general, IT-community to make several steps ahead, but… remaining far beyond the prophecies of science and fiction writers addressed to the beginning of 21st century. At the same time, we had already been accustomed that engineering achievements overcome even the wildest predictions of writers. So one can obtain the feeling, that, probably, all is not so fair in the Dutch Kingdom… Wise people say that if you encounter too serious obstacles on your way – think over your directions and methods once more.

So I was brought to the feeling that the time has come to reach better understanding of the subject which we are going to simulate and of the social activity processes with which AI creatures have to naturally interact [Lozovskiy, 1990-3, 1992, 2001]. The other motivation for the current research was the spell of esotericism, including "magic", extrasensory perception, non-traditional healing. It could happen that probably we even do not possess enough understanding how to use our natural human possibilities in handling knowledge, conscious and subconscious sphere. These considerations attracted my attention to problems of noosphere, religions, magic, faith, soul, and even God and Absolute [Lozovskiy, 2003, 2005]… Further investigations until now did not show any proofs of anything "supernatural" in our life and Nature. But the outstanding role of psychology and socio-psychology were understood to be of primary importance in our life.

More than that… One can argue that the efforts in AI development ignoring these aspects and information directed natural evolution cannot be winning. This research, of course, is only at its outline stage, but going will master road.

Physical, Mental and Cultural Domains of the Noosphere

AI started as an effort to simulate consciousness and behavior of a human being. The time has come to revise the object and the environment of our attention. First of all, our comprehension of the World became more balanced. Now we understand that, in the noosphere era we should adequately consider and simulate three types of entities: physical, mental (conscious) and cultural (social) [Lozovskiy, 2003]. Physical ones reside in the objective world – buildings, rivers, cars, live beings as physical bodies, galaxies, physical fields, energy, et al. Mental – are inhabitants of the conscious sphere of each living human: one's knowledge of mathematics, nostalgia for the birthplace, feelings, emotions, attitude to others… Cultural sphere (ideosphere) comprises all sciences, arts, fine arts, customs, political and economic systems, beliefs, religions, know-how, the adopted notions of purely «human» relations and feelings (fear, hatred, love, trust, admiration, amazement, irony, etc.) created and accumulated during evolution.

From the other viewpoint, we should understand that human consciousness could not be studied and simulated as an isolated phenomenon. Isolation creates Mowgli… Human culture, language, interaction with noosphere cannot be thought about without due consideration of psychology and socio-psychology of interactions in humans. This sphere, or dimension, of human existence is of paramount importance in intellect and consciousness development. One can be almost sure that underestimating this aspect can bring down the hope for AI creation.

Gone with the wind are efforts to build AI system on the basis of theorem proving. Afterwards it was found that human reasoning in last turn reminds theorem proving. It is, first of all, based on subjective apprehension of reality and on personal goals which are to be achieved. That is why we are witnesses of disputes between subjects and existence of various viewpoints. This domain is almost virgin for knowledge engineers. The situation aggravates due the enormous role of psychological and socio-psychological aspects and inner springs in human behavior. Here I will try to briefly consider only some of these issues.

Information within Material World

My personal interest in this theme started from esotericism. The first impression was that AI specialists probably ignore the great bulk of functional power of human mind and behavior. Problems of faith, belief, soul, God, magic had to be studied more closely, giving constructive definitions, exploring the real power of these theories and techniques.

Earlier [Lozovskiy, 2005] I argued that materialistic world-view can pretend for better adequacy to reality, than the idealistic one. Ideal entities do exist in reality, they "feel" themselves quite natural and "cozy" within their material frames, but their status is somewhat more delicate and complicated than that of material ones. Consistent materialist philosophy approach precludes us from the fallacy of considering matter, energy and information as having equal rights as fundamental World-nature building bricks. In order to put this issue straight, we need to reconsider the notion of signal, information, knowledge, interpretation and intellectual subject.

Material (physical) world consists of substance and energy. It is populated with objects, which interact in some or other way. Interaction occurs according to certain objective Laws (properties) of Nature. We shall not argue now, who "created" this World including its Laws and "put it into operation", how it sprang into existence. What I would like to emphasize now is the "law-abidingness", naturalness of all processes occurring in this world. The Laws of Nature have the general cause-effect form. They are not spontaneous and do not change sporadically from time to time. Materialism denies existence of any consciousness, goal directed activity at this level. During its history, the Earth passed through the epoch of cosmogenesis, geogenesis, biogenesis, sociogenesis (civilization), and is now at the stage of noogenesis [Lozovskiy, 2003]. Our Universe now, besides the physical world, comprises mental worlds of currently living conscious beings and the cultural world of the whole humanity. In order to consider problems of knowledge, information, cultural stratum of the noosphere creation, we should analyze the process of knowledge acquiring by intellectual subject (IS) through its sensors, and then – information exchange between IS'es.

The conscious of live being receives signals from its organs, body parts and from environment through its five (?) senses. The sources of these signals are purely physical: movement of some objects, matter or flows of some energy. We shall call them primary effects (PE). Sometimes certain instrumentation is used to change modality of these PE, transform them to the form which could be perceived by the sensors of live being. Humans use microscopes, telescopes, thermometers, radio signals receivers, x-ray indicators, etc. The raw PE have limited usability for the living creature. Their receptor cells, organs, senses and nerve system assemble them into primary patterns, sometimes using preprocessing, detecting certain features $\{f_1 \ldots f_n\}$ potentially useful in being's goal directed activity. These primary patterns are encoded into certain material media in the form of signs. A sign *S* here is an entity which signifies these features. Usually there is a set of signs signifying different instantiations of features: $S = \{S_i\} = \{f_{i1} \ldots f_{in}\}$, where $i = 1, \ldots m$ – specific instantiation of *n* features. *S* we shall call an alphabet of separate signs S_i . So, the main reason behind signs is that they signify some prototype entities – denotations. They say that the denotative value of the certain sign is specific denotation. This correspondence remains implicit. And it is the root of the notion of intellect, consciousness. Signs are the simplest ideal objects in the noosphere. Their creation and usage is closely entangled with the interpretation activity of conscious beings. The

basic process behind it is reflection, mapping of reality into the knowledge sphere of the conscious beings. It is done *informally*, basing on the ***personal*** experience of the being.

Thus, we are speaking now about subjective mental signs, i.e. signs in the conscious sphere of the specific IS: $S^m = \{D, Sg, IS\}$. The mental sign is a triplet: specific denotation D, its designation Sg and IS itself in whose consciousness and from the viewpoint of whose goal directed activities this mental sign reside.

Of course, these interpretations are somehow correlated between live beings in the process of their social interaction giving some cultural representation – common denomination. “1:1 In the beginning was the Word, and the Word was with God, and the Word was God” [Bible]. That is how the human culture started – with the words of language, with signs, which were used to explicate mental signs of people socializing them for human communities, forming the cultural strata of the noosphere. Of course, by “word” we understand here not only words of the natural languages, but also the pictures, ritual dances, cave art – all situations of signs creation and their explication in the form of such or other signals.

In order to transmit signs to the proper destination signs themselves – ideal entities - cannot be used. Instead are used signals – material objects representing, substituting signs. The signals are created *formally* according to the laws and rules exploited in the process of sensors engineering (if they are technical devices) or in the process of evolution, if we speak about senses of live being. Communication networks, through which signals are transmitted, handle them formally as pure material objects. They, of course, do not “understand” their meaning or value in any way.

Let us consider now the conscious being – communicant, which receive some signals from its senses. Its first task is to interpret this signal reconstructing the source sign, i.e. the meaning in the terms of its denotation. Interpretation results in obtaining the set of parameters – attributes which specify corresponding features extracted by senses at the stage of interaction with reality. Usually these set of attributes are considered as connotative semantics of the given sign – its informational content. The second task is to match this sign against its own knowledge and data base (KDB, or simply: knowledge). At this stage the received information is evaluated.

Creation of the SELF identity. One of the first problems to be solved by the conscious being is to segregate itself from environment constructing its identity – its SELF. In this procedure exceptionally important role play own activity supported by being’s effectors. Here the physical body of the being becomes the research and tuning instrument. For example, a kid sees a bright toy hanged over his/her cradle. First he just looks at it. Then he stretches a hand and touches it. This process is not straightforward – the kid at first cannot move his hand directly to its destination. But the goal of the movement is evolved, and the kid tries to accomplish it. When he sees that his hand moves in the wrong direction, he acts with his muscles to correct its trajectory. Even during the process of solving such primitive task, the kid learns to discern feelings obtained from his limbs and from objects in his environment: they are “closer” to him, and over them he has better control. Besides, he teaches himself to construct sensor patterns of possibly different sensors. In our example with touching a toy – he combines the sensors of arm and hand position, acceleration, fingers’ control, touching the toy. At the same time he becomes acquainted with the objects belonging to its environment, which he can perceive, but his control over them is limited. On the next stages the space around the subject is explored, the measure of proximity to SELF is studied empirically.

This procedure of new identity creation is performed by any driver, pilot, bicyclist, even the typist in the process of mastering their engagement. After the certain period of coordinating receptor patterns and effecter control the subject obtains the feeling akin his body expansion, becoming a whole with his apparatus. Even the apprehension of geometrical dimensions in the consciousness of the driver changes so that he on the reflex level controls the driver + his car complex rationally during maneuvering.

It is instructive to evaluate information-delta – the state change of information within the KDB. Five alternatives are possible here.

1. ***Deferred or irrelevant information***. It happens when correlation with being's KDB fails – being cannot "understand" the sign received. So, it could be stored for possible future consideration, or just rejected if such procedure is not planned. The value of received information (KDB knowledge-delta) at the time being is nil.
2. ***Redundant information***. Information completely duplicates some fragment of the KDB. It can be rejected as useless, and the value of knowledge-delta here is again nil.
3. ***Information augmentation***. This happens when received information correlates with the current status of the DBMS and can be used for its augmentation, bringing something "new" about the problem domain. Knowledge-delta here is positive.
4. ***Contradictory information***. Received information can be correlated against the current state of the KDB, but it contradicts with information already present. In this case, there must be initiated the procedure which should investigate this case and decide who is who here. Knowledge-delta is nil, if the new information will be classified as erroneous, and positive if it will lead to bringing the KDB in better correspondence with reality.
5. ***Erroneous information***. Formally it appears to be augmentative information, but in reality it erroneously reflects the situation in the problem domain. Knowledge-delta here should be considered negative.

The aggravating factor at this stage is that frequently the information received cannot be correctly classified on the fly according to types just presented.

Up to this point we have considered the process of signs creation within some conscious being. These signs are based on its personal senses (PS) - sight, hearing, touch, smell, taste, transformed into signals, which reach its sphere of consciousness and as a result of signal interpretation become information chunks which can influence (augment, decrement or modify) the KDB of this being. This knowledge remains within the consciousness sphere of this IS. But the cultural sphere support requires explicating this knowledge and passing it there. Only this process can make personal achievements of one being to become a quota in the treasury of all-human knowledge.

There exist two methods of passing our information to others: direct (personal) and indirect (via the cultural noosphere domain). Everyone will agree that personal is more efficient. And it is not only the due convenience of multimodal broadband communication between the subjects. Very important is that we frequently can use the direct method of passing signs to our collocutors: I show a tree, and say – it is a pine. Everyone can come up, see it, touch it, smell it, feel its beauty… That is why, when we cannot show to our pupils natural object, we show a picture, draw it ourselves on the sheet of paper – trying to approach to direct situation as close, as is possible.

But when there is no means of passing information to our communicants directly, for example, to our progenies – we use various methods to fix it in some form for future use by anybody to whom it may concern [Lozovskiy, 2005]. Thus the cultural domain – ideosphere - is formed and nourished.

The consideration just undertaken helps us to formulate the following definitions.

Intellectual subject (***IS***) is the being which can receive signals from its senses or from other IS, interpret them obtaining information, knowledge, ideal objects which it uses in its goal directed activity.

Sign is the entity created in the KDB of some IS having its denotation in the problem domain and designation - in the KDB of this IS supported with the connotative semantics needed for the goal directed activity of this IS.

Signal is the symbolic representation of some sign using specific material basis, carrier. Signals are produced by IS in the process of signs' encoding, modulation of the carrier. Examples: books, acoustic speech, radio station transmission.

IS willing to obtain some information from the other IS directly or from the KDB of cultural domain should have the possibility to decode signals from these sources and interpret them according to alphabet, vocabulary and language rules used. Of course, information obtained should be relevant (at least, potentially relevant) to the success of goal-directed activity, in which this IS could be involved.

Let us summarize the main issues of this section from the viewpoint of the leading AI objective: solving the enigma of human consciousness origination.

1. At the dawn of intellect appeared signs: special constructs comprising denotation in the problem domain, its connotation in the KDB of the being which was done for the benefit of some type of its goal directed activity.
2. Joint functioning of subject's receptors and effectors led to segregation of SELF identity out of the environment. This step is the indispensable requisite for behavior planning of the being.
3. IS had to communicate between themselves and create the KDB of the humankind – which gave rise to the ideosphere: cultural domain of the noosphere. Communication was done by means of signals transmission, where signs (ideal objects, chunks of information) were first encoded, attached to the material carrier in the form of signs and then due the inverse procedure – interpretation – ideal objects (signs) were reconstructed.

My conjecture is that this procedure, in its integrity, is vitally important for consciousness and intellect birth either in human society or in the course of this phenomenon simulation. It is emergence of the soul and spiritual life.

Beliefs as a Basis for Knowledge

I think belief is the notion of primary importance in knowledge engineering, psychology and in esotericism [Lozovskiy, 2005]. Here we shall study further elaboration of this theme.

Belief is a situation when model entity in a knowledge and data base (KDB) system is considered to be in adequate correspondence with prototype entity in given problem domain (PD). Strictly speaking, all model entities in any KDB are beliefs. Beliefs can be classified from the viewpoint of pragmatics.

- ***Axiomatic beliefs in formal systems.*** These systems comprise axioms and inference rules. Axioms are assumptions taken by IS without proof – on some pragmatic or aesthetic foundation, for example, axioms of Euclidean geometry. They are true on assumption base, i.e. for all those IS who agrees to believe in them. The characteristic feature of these beliefs is that they are completely abstract and need not have denotation in the PD. So they do not intersect in any way with the "real" life and are in no way "responsible" for any discrepancies between "the theory" and realities of PD.
- ***Hypotheses in formal systems***. If we accept some axioms to be true (axiomatic belief), so, on the formal basis, are true all correct inferences from the axiom set – theorems, which supplement the axiomatic basis of the given theory and could be used for future inferences. Hypothesis in some formal system is the certain statement, which given IS lets, or assumes, to be true – ***believes in its validity***. This statement remains hypothetic until it is proved or disproved within the frames of the given formal system. Proved hypothesis becomes a theorem.
- ***Natural-science beliefs, or AD-hypotheses***. Probably, they belong to most complicated category of beliefs. Researchers formulate certain assertions, or systems of assertions concerning the realities of the application domain (AD) considered. First, they have the status of hypotheses, i.e. beliefs. Eventually

these hypotheses should be verified: theoretic, or model, facts or estimations should be in specified correspondence with known real data – according to specified requirements and accuracy. Besides, AD-hypotheses are tested in the prediction mode: the inferences about unknown, or future, AD facts should be confirmed by the factual information during future AD studies. Usual requirement to these confirmations is stability obtained irrelevantly of factors, which should not influence measurements: invariance to time, place, instruments, persons, et al. All these requirements should be scrupulously specified by the author of the hypothesis. While more confirmations of this type are obtained, more plausible the given AD-hypothesis is considered. While testing of a hypothesis supplies exclusively positive results, its weight increases, and gradually it drifts towards the "theory" qualification. But researchers should not loose wariness and look for falsifications of such "theories". One reliable falsification is enough for reconsideration of the theory – either slight, or even principal. Sometimes, it leads to its complete abandonment. Thus, "reliability" of natural science "theories" can vary in the broad range only asymptotically approaching the 100% figure.

- ***Extrapolatory AD beliefs.*** These beliefs are variations of AD-hypotheses and apply to events, processes or parameters' values, which, according to given IS understanding of phenomenon considered, should happen, or actualize in future. This belief may be based on intuitive extrapolation of observed process. Not excluded is the use of algorithmic methods. It is important to notice that it is impossible to accurately predict future on the basis of formal methods: processes in the AD are always liable to uncontrolled influences and restricted accuracy. According to [Ashby], in AD we always have to deal with incompletely observable systems. Such systems, from time, to time behave themselves whimsically due the interference of unknown to us factors. It is not the intrinsic statistical essentiality of the given phenomenon – it is just limitation of our knowledge.
- ***Belief-insight***. Sometimes humans are embarrassed to explain their arrival to some idea. Frequently it happens quite suddenly without noticeable process of reasoning and inference. Mystics explain it with the help of God's providence and direction. At the same time they slyly omit the fact that such revelations condescend only to specialists in specific field of knowledge, who spent much time and efforts studying their theme and trying to arrange their knowledge in some systematic form. Of course, the great support comes from one's subconscious sphere.
- ***Cultural (ideosphere) beliefs.*** The human progress brought us the noosphere phenomenon with its ideosphere including ideal entities, created by humankind: sciences, arts, religions, etiquette, folklore, habits, customs, etc. Every human belongs to the certain historic, social and culture stratum, accepts linguistic, ethic and aesthetic habits and conventions of his native land. Holding to specific cultural beliefs tends to consolidate people within corresponding social units. Existence of cultural beliefs is indispensable feature of human progress. Everybody accepts, as an initial capital, achievements and inventions of previous generations. He should assimilate them, believe in them and add there his own achievements. Without acceptance and assimilation of these beliefs, each human being would have to start his own progress from the zero level and all his knowledge accumulated during his life will vanish with him. It is clear that a person cannot check personally all the bulk of knowledge produced by civilization: he has just to believe in it relying on the joint human experience. Of course, sometimes even the conventional truths are reconsidered. But essentials of this notion – cultural beliefs – remain. There are several kinds of cultural beliefs. In what follows we shall consider absolute belief, mystic theories, confidence and egregors.
- ***Absolute belief: faith.*** This belief usually concerns Intangible World, about which, according to the definition, we have no positive knowledge. Faith, religious belief in its purest form does not need

motivations, argumentations and confirmations. It is given to its followers “as is”. At most, you will receive reference to some “Holy” books or to authority of certain “known” people. Pure beliefs have nothing in common with World comprehension and can be thought of as some variety of cultural beliefs. Usually specific faith is adopted within certain community and is presented to children as indispensable attribute of their life: “You should believe, because all our people believe”. In folklore, fiction literature we have popular genre of fairy tails, which are close relatives of religious beliefs being the product of people’s dreams about better world, or having some didactic moral.

- ***Mystic theories (MT).*** They closely resemble absolute beliefs. From the other side, the term “theories” is used here intentionally: creators of MT present them as if they are natural-science theories, which explain our World, Nature, Universe, human behavior… By the way, the term “theories” is, of course, exaggerating here – even to use the term “hypothesis” there should be certain grounds. Main efforts are put there to perform this mimicry: you will find numerous references to natural events, to social situations; even situations in human life, human feelings, psychological state, emotions, in relations to other persons – everything suggestively “illustrate” the adequateness of such “mystic theory”. In argumentation you frequently find phrases: “It is well known…”, “Scientists have proved…”, “Numerous evidences exist…”, - but you will never find exact clues to any of the mentioned “facts” or documents. As a support, very popular are the references to prominent, well known authors of the past. Sometimes they name contemporary authors, which, of course, are academics of certain self-made academies having no relationship with National Academy of science, but this nuance is frequently omitted. The funniest thing is – when you pick such source materials – you will frequently find there the same rootless argumentations. The trick is in the hope that many readers will believe in the given “theory” and will not check the issue from the sources… You will never find such thing as mystic hypothesis – only “theories” which sprang into existence by some irrational way – no one will answer you the questions: “Why do you say so?”, “What validating facts exist in its support?” Frequently you will be informed that on the basis of the given “theory” many useful devices with fantastic capabilities were implemented. Of course, you will never obtain factual confirmations for these claims. Good analysis of mysticism and MT was performed by [Nan]. In the similar sense he uses the term “virtual patterns”. Frequently MT use very dim undefined entities, for example – the concept of God, spirit. These theories are usually very general – they explain ***everything***... Of course, they are completely independent from natural-science and other MT. Usually these theories rely heavily on human psychology and socio-psychology making very far-reaching conclusions about general principles of World construction, problems of matter, energy and information. MT is characterized by absolute isolation from reality. On the basis of such theories one cannot perform certain actions with guaranteed result. Sometimes something “works”, sometimes – not. Experiments typical for natural science are rejected by MT adepts from the doorstep. Frequently MT is declared as completely subjective one – that is why you cannot ask somebody to demonstrate practical results: you should work out and believe in them yourself. MT closely resembles religion, absolute belief. Sometimes certain practical effects are used in support of specific MT. The closer analysis show that they are, in fact, independent: the same effects could be achieved on the basis of some other MT or without any theory at all.

- ***Personified, authoritative, social belief, or trust***. Trust has deep roots in cultural belief and is based on personal relations of believer to other person or other source of knowledge. If I trust to somebody or some knowledge source – I will believe in facts and advices, which I obtain from this source. Children trust their parents, students trust in their teachers, absolute believers trust in their Gods and prophets. Of course, we can argue about foundations of such trusts. Sometimes we trust to sources, which proved their reliability in the past – personally or transitively - due advice of somebody in whom we trust. Sometimes – because we

should or have to. Regretfully, uncritical social belief frequently leads people to belief in mystic theories. Being asked – why do you believe in this – the answer is: because He (?) said it…

- ***Autosuggestion.*** Using computer slang, this belief could be named self-programming. These beliefs are constructed in one's own mental sphere. Belief in victory, in overcoming obstacles, in own recovery – they all program our conscious and subconscious spheres for insistent goal-directed activity, helps our personal neuro-somatic system in adjusting functional disorders, stimulating our activity in the specified direction. "One but fiery passion" is the powerful mechanism for making one's dreams come true.

Concluding this section dedicated to various kinds of beliefs, we can state the following.

1. Beliefs are indispensable part of our conscious goal-directed activity.
2. Any belief should be crowned by some important ***real*** outcome, must be proved to be true: "the proof of the pudding is in the eating". This outcome not necessarily is material – quite possible, it will manifest itself in some ideal, psychological form, but it has to do it!

Egregors and the Cultural Noosphere World

Egregors [Egregors, 1-4], [Bernstein] are probably among the most powerful and distinguished components of our culture. They are both its creature, content and indispensable part. More than that – without egregors neither existence, nor development of our noosphere cultural strata could be possible. Here is my definition of egregors.

Egregor is the system of socio-psychological relations forming certain social unity with the set of some common features: language, goals, interests and other bylaws. Examples are: nations, countries, organizations, societies, teams, families, religions, political parties, affinity groups, sporting or hobby circles, various geographical or situational communities (population of a town, village, district, current passengers of a given bus, participants of some birthday party et al).

Each person belongs to several egregors at a time; some being rather stable, others quite situational. He cannot be "free" – he is bound by rules, written and unwritten laws, regulations, moral "do's" and "don't's". So, his behavior is a resultant force in this complicated informal superposition of different egregorial tendencies and influences. Understanding human behavior from any point of view: societal management, governmental policy, simulation of intelligence and organizational management systems creation is impossible without closer acquaintance with egregorial technique, their internal structure and dynamics.

Determinative features of egregors are issues of their content, interrelations between their members, relations between egregor and its members, and "outer" relations: with other egregors and their members. Egregors manifest themselves in powerful factors of psychological influence and inter-influence on conscious and subconscious levels, which frequently add to them certain mystic hue. Theosophists associate with egregors fine-material fields, energy or entities, which can "suck energy" from participants, protecting them, in their turn, from (hostile?) influence of other egregors. Up to now I have no confirmations of any external (besides participant's psychics) existence and properties of this mystic energy-information egregor component. Their social and psychological components really have great influence on people. We shall study this issue closer.

- ***Existence of some common grounds***. Specific community is created around some idea, belief, sphere of common interests, or just on the basis of certain common features of their members: territorial proximity, geopolitical preferences, relatives, graduates of some university, common language, ethnicity, hobby, religion, sporting teams, organization's staff, et al.
- ***Teaching***. The most powerful and durable egregors have certain Teaching (from capital "T") in their core. It plays the role of the central kernel, bearing axis which allow put the Movement, societal mechanisms to the full swing. The Teaching should bear some great spiritual idea – achieving total happiness, world of

general welfare and justice, redemption of sins, self-improvement, achieving Kingdom of Light, eternal life, merging the God. All known religious cults, socio-political movements are here: socialism, communism, fascism, nationalism, chauvinism. Very important is the Teaching's age. Antique teachings accrete with own myths, history (real or fictitious). These factors render significant influence on their participants' psychics.

- ***Originator of the Teaching***. Weight of the Teaching is effectively supported by the personality of its founder, originator – either real or mythical, his companions and disciples. As a rule, the founder's and disciples' portraits are delivered. They are necessarily idealized, taken at the earlier age, intensively retouched. Grey hair and other chevelure deficiencies usually are repaired. Worshiped personality is ennobled, natural facial asymmetry – corrected, frequently kind smile enlightens the face, or, at least, eyes become more affable than in reality, height is corrected. Sometimes, the leader's images become medal-like. Compositions can be used, where leaders are put together with the jubilating crowd in the background, rays of the raising sun, sky, white clouds... One should pay special attention to the personality of the current leader, or the head of the organization. If he, at the same time, is the founder of the Teaching or organization, if he pretends for special distinction: being incarnation of God, Jesus, Buddha, modern prophet, superhuman, extraterrestrial intellect envoy, if he does not share his lordship with nobody else, pays the primary attention to boosting his personal authority – we may deal here with a sect, probably, destructive cult. Instead of explicit (implicit are always present) leader, some other object of worship could be present – fetish, artifact, deity, natural power, et al.
- ***Documentation***. In the foundation of egregorial formations usually there is documentation concerning this community or Teaching. It could be some "sacred book", or books – in case of religion, unwritten laws, ethnic traditions, statute of the party, marriage contract. "Sacred books" usually include abundance of (different) elucidations, explanations and interpretations. Frequently you find there evidently contradicting items: it helps sacred book interpreters to justify any politically alleged action. Sometimes simplified versions are provided for children – using understandable examples. Children, women, youths are approached with the keen attention. Within youths' programs are being thoroughly developed sublimed ideals, positive patterns, various parables, stories with educative undertones. Most powerful, influential egregors are attraction points for various art workers: painters, litterateurs, sculptors, composers, which, according to their talent and capabilities, produce emotional and iconic background for the Teaching they back up. By the way, the quality of such humanitarian production frequently can be outstanding. These artists' authority and immediate influence on people's psychics efficiently support their political order.
- ***Membership regulations***. Special regulations may exist concerning the membership. Officially registered egregors, as a rule, have regulations, where the principal ideas of the corresponding Teaching are formulated together with mandatory requirements to the persons allying the Society. Becoming a member may be free or paid, an applicant should satisfy some requirements (believe in The Idea, have certain qualities, or be eager to obtain them, etc). So, there exists a procedure of initiation, enrollment into the community. Special rituals of joining the community are developed. There can be several qualities of participation according to candidate's maturity, preparedness and merits. Rites for initiation, consecration, rank granting are worked out. They can be simplified, constituting, probably, only the informal talk; sometimes, written or even oral statement is enough. But it can be a pompous ritual, oaths with summons to the party's functionary, mystic ordinance in the presence of consecrate, senior colleagues. The neophyte can be exposed to certain trials, exams, to which he prepares himself in advance and agitates together with his relatives and friends. Analogously, formal procedures and rituals exist for condemnation of member's misconduct – including exemption, disfellowship or even the more rigid punishment.

- ***Member's rights and obligations***. As a rule, these documents are an inalienable part of the documentation corpus of organization. They may be written, or be words of mouth and fix members' obligations and rights – in material and spiritual aspects, their do's and don't's, how to treat their egregor's outsiders. Each member has a set of liabilities concerning the broad spectrum of what should and what should not be thought, said and done at proper circumstances. As a reward community's member receive sometimes material, but – what is much more important – moral, spiritual support, that wonderful feeling of participating in some great, true, even holy activity. Frequently, such community promises take a form of future happiness to come, maybe even after physical death. This issue has paramount importance and is probably determinative in the whole community–members' affair. This mechanism works on psychological level and has powerful suggestive effect. They are suggested that they are participial to the great and sacred affair. Pictures of happiness and glory are depicted, overall happiness, sublimation and awards, usually, in future, or even in future life… In the later case, the conception of postmortem existence is developed, where generously all possible rewards are promised: sins forgiveness, piety and other feats in this life requital. Special attention is paid to mutual support of egregor's members. It could be both moral and material – money, roof over head, kind attitude, caress, empathy, spiritual support – up to trance, suggestion methods and psychedelics. As a result, community member arrives at the conviction that he has found true defenders and mates, which surely will punish his offenders and rescue him from dangerous situations. Members of egregorial community frequently use specific addressing towards each other: friend, comrade, colleague, brother (sister) et al. All these factors, in complex, have exceedingly strong influence on the psychics of each member. He becomes overwhelmed with "energy", sincere and cheerful feelings of belonging to something great, glorious and honorable, and it creates so strong ties with the Teaching, he appreciates his membership even more and more, and, according to constructed positive feedback, with all his might yearns to become better, escape blaming and punishment, longing for praise and acknowledgement of "senior comrades". Personality suppression methods are widely used, which make possible members' behavior monitoring and management. Resulting human beings sometimes are easily recognized due the unhealthy eyes glister, excess activity, or contrariwise – excess aptitude to depression, illegitimacy of behavior, failure to communicate in themes diverging from those adopted in his society.

- ***Membership certificates***. Sometimes members of a community are formally registered and receive personal certificates confirming their membership. It adds to their confidence and self-respect, impresses relatives and acquaintances. Sometimes such certificates are used to obtain certain benefits or preferences beyond the societal boundaries. When putting inadvertently such certificate into somebody's face, its owner really feels the power of his egregor behind his shoulders. Similar sensations are expected (or really induced) in his opponent. Psychological, suggestive power of an egregor increases with its age and the number of its members. Egregor's heads and officers adore mentioning these figures, and also other ones, characterizing egregor's power and penetration into society.

- ***Symbolism and rituals***. It is the last in the list, but not less important factor aimed at egregor's unity, power and increase of its influence on others. Usually, special thoroughly developed symbolism is present. Official blazon, colors, uniform, flags and banners, badges, ribbons and knots decorating suits, dresses, car antennas et al. The same color gamut is used for scarves, ties, bras, slips and bathing suits. Sometimes this stuff is distributed free of charge or for just symbolic price – so that even plain people can demonstrate their support of new power. Special garments fashion can be introduced, which is carefully thought over with attraction of good designers and artists. It also had to discipline members, render impression on others, suggesting respect and prompting them to join the Society. Considerable attention

is devoted to acoustic impacts. Besides the acknowledged pieces of music and choreography: songs, marches, hymns, dances, - widely used are NLP (neuro-linguistic programming) and various hypnotic, suggestive and trance techniques – including drums, which role could play even empty iron barrels, fifes. Songs also comply with strict social control task; widely used are primitive slogans, short words, affirmations, their multiple repetitions. All these sounds, at the pain threshold levels, conforming with the certain rhythm aimed at psychic's suppression suggesting inevitability of the coming… no, already achieved victory of the new Teaching, new bright and just social formation.

What can be said as an implication of this rather lengthy consideration?

1. The phenomenon of egregors is total: our life in its social aspect is superposition of oodles of egregors. Each one is organized according to definite principles, but in their dynamics they interact in rather complex way producing conflicts and resource clinches. That is why planning goal-directed activity in AI remains so difficult task in real applications.
2. The main clue to understanding this aspect of human behavior is further investigations in the sphere of human psychology and sociology. It is inevitable from positions of human society management enhancement, and as a countermeasure against evil attacks from antihuman forces: destructive religions and cults, imported "color-revolutions", attempts to subdue or zombify people.
3. In the course of this research effort was not revealed a single argument in favor of any supernatural, "fine-material" foundation which according to some esoteric sources is an immanent component of egregors. No mystic influences, no mysterious "magic" – just psychology, though much more elaborated, than "classic" one.

Magic. Psychological Hypostasis

Miracle for certain IS is some activity, or result of an activity, when it (she, he) fails to follow the cause-effect dependability between events. According to [Augustine], - miracles are not contrary to nature, but only contrary to what we know about nature. Of course, this definition is really subjective, dependent on IS, its knowledge base, attention and goals. What is miracle for one could be quite evident for the other. A miracle can vanish, if somebody explains you the situation.

Magic is a miracle demonstrated by some IS, which is called ***magician***. Pay attention that miracle demonstrated by non-IS remains miracle – not magic.

Juggler is a magician, who demonstrate "self-made' miracle created by knowingly concealing or distorting some cause-effect structure in the process of his performance. Thus, we can say that magician is "true magician", while juggler is only pretending to seem a "true magician".

The nature of a miracle. The phenomenon of a miracle is determinative for the process of World cognition by civilization (system of IS'es). Research efforts stem from revealing discrepancies between expected course of events and observed one, that is, first, we reveal some miracle, and then discrown, demiraclify, demystify it by working out explanations, models, theories, concepts, even new paradigms – if it is needed. Thus, science functions as destroyer, extinguisher of miracles. Here is the main controversy between scientists and humanitarians, who prefer to "play" with miracles, live in mystic worlds. Yes, existence of miracles spiritualizes our life. Unanswered questions should always remain the part of our culture. While scientists demiraclify the World, humanitarians defamiliarize [Defamiliarization] it. Here is dialectic contradiction in action. Disappearance of the last miracle would be the death of the culture.

When some IS encounters a miracle, we can put a question about its explanations: are they rooted in the problem domain, in his cognizing world system (psychics), or both. For every IS the whole Universe can be divided into

two parts: its internal, subjective, psychic, mental [Lozovskiy, 2003] sphere and all the rest material World. Of course, our IS is not an isolated being – it exchange signals with ideosphere, with other IS's and transforms them into knowledge thus nourishing its own mental resource, supplying, in his turn, his own contribution to the all-human noosphere.

I shall not pay special attention here to the material World. It ought to be done by specialists in physics, biology, astronomy, et al. according to classical natural-science approach: gathering of evidences - hypothesis formation – experimental validation while checking invariance over instrument, experimenter, time, place change, predictions, and accumulation of reliable statistics – lookup for falsifications – hypothesis adjustment, or even paradigm change – recurrence to experiments. During these iterations hypotheses gradually earns the status of natural-science theory.

My conjecture is that majority of mysterious effects which overwhelm esoteric books has psychological basis. The main principle here should be that of Occam's razor: The principle states that the explanation of any phenomenon should make as few assumptions as possible, eliminating, or "shaving off," those that make no difference in the observable predictions of the explanatory hypothesis or theory. In our case, if we have evident materialistic explanation of some phenomenon, there is no need in alluding to any mystic, fine-material and the like esoteric factors. In esotericism the antithesis to Occam's razor principle – that of the suslik is very popular: "Do you see suslik? - No... - Neither do I!... But he is there!" (film "ДМБ"). Hereinafter we consider several "esoteric" phenomena.

First of all, I should state that all my efforts from 2003 up to now did not produced reliable evidences of any objective "supernatural" effects in the physical world. Neither telepathy, nor clairvoyance, telekinesis, levitation… The healers and "clairvoyants", with whom I contacted, dealt only with humans. They either strongly opposed to participate in pure statistically reliable experiments, for example, Zener cards reading, or results obtained were within statistical expectancy limits. The resulting natural impression was that probably one should carefully study the peculiarities of human being as an object of esoteric activity. The most important, discriminatory feature of humans is their consciousness, fine psychological processes, accompanying all intellectual activity, including social psychology which is the backbone of all egregors. The phenomenon of consciousness, goal-directed behavior is the result of gestalt-effect on the basis of the material entity – human being with all its structures, subsystems and processes, organized "in proper way". Performing the inverse operation, - that of analysis - we inevitable find psychology as the determinative hypostasis for behavior of humans.

Magical impacts: malefice, spiritual injury, damnation. We shall not go into fine distinctions of these acts, as superstitious people do. In what follows, I will try to nominate several important aspects, common to all known to me magical acts.

1. All humans, before they can experience any magical influences, should be prepared. If we deal with religious influence, they should know the main dogmata of this religion, to be believers, be included into the corresponding egregor. In case of orthodoxy, they must believe in God, Jesus, angels, paradise, hell, Satan, sinners being tortured in the kettles with boiling pitch. In esotericism very popular is the notion of "energetic entity", which is – as they say – some fine-material being, which could suck energy from the poor victim, or implant some malicious "programs" into his soul, changing his behavior in malicious way… Superstitious people afraid of black cats, number # 13, Fridays, etc.
2. If the given person is not yet "prepared", the primary magician's task is to "explain" him what he should be ***afraid of***… Please, mention, that one of the most efficient instruments in magic is a ***fear***, better – terror. Magician, in order to be successful, has to rip out his subject from the adequate nature perception, replant him in the field of magic theory, where precedently adequate thinker becomes the true believer.

3. To the full swing is used suggestion – in every possible form: classic, Ericksonian, Gypsy hypnosis, NLP.
4. Powerful influence is experienced by a subject participating in deeply and broadly developed egregors. This influence can be so intense, that mystics attribute it to some supernatural forces, immaterial energy and information field.

Virtual communities – forums, chats and trolling. The progress in Internet communications facilitated creation of virtual communities – specific form of egregors, where participants can be spread geographically all over the globe, but they are bound with some sort of common interests, science, technology, hobbies, language, and way of life or behavior. Each participant has the broad spectrum of features to create an image he likes. He can change his real age, gender, appearance, use nicks and specific pictures – avatars – representing his identity. It opens the door to real identity proliferation. Thus one physical person can develop several absolutely distinct personalities. Changing your virtual identity gives phenomenal experience absolutely unachievable in real life. For example, very popular become identity deception games. These virtual communities become more and more efficient substitute for the "real" physical contacts. Of course, psychiatrists should be aware and preclude our society change to fantastic one, foreshown by [Aldani]. Forums and chats became widely accepted practice today.

A few words should be probably said about [Trolling] – less known practice. Trolls can be existing members of a community that rarely post and often contribute no useful information to the thread, but instead make argumentative posts in an attempt to discredit another person, concentrating almost exclusively on facts irrelevant to the point of the conversation, with the intent of provoking a reaction from others. A troll is a person who approaches a board with the specific intention of stirring things up, either with no particular motive or provocation in mind, other than to be purely destructive or if the motive or provocation is against the ethos of the board. A troll can disrupt the discussion on a newsgroup, disseminate bad advice, and damage the feeling of trust in the newsgroup community. Sometimes it can appear as the Devil's advocate.

Good and bad luck. Sometimes, people become fascinated by the overall arrangement of their life attributing it to their "fate", destiny, karma, or just saying that somebody is lucky, or, contrariwise, pursued by bad luck. Esotericism pretends to give certain mystic, transcendental explanations, while quite straightforward explanation is given by inherent and trained personality type plus evident practice in memory and attention development, positive attitude towards neighbors, meditation, relaxation and concentration practices.

Premonition. Sometimes, we can hear stories about premonition, forecasts, anticipation of some events. These phenomena are also attributed to Providence, or some extrasensory connection with global energy-information field, **Akashic records** and the like. The Akashic record is an imagined spiritual realm, supposedly holding a record of all events, actions, thoughts and feelings that have ever occurred or will ever occur. Theosophists believe that the akasha is an "astral light" containing occult records which spiritual beings can perceive by their special "astral senses" and "astral bodies". Clairvoyance, spiritual insight, prophecy and many other untestable metaphysical and religious notions are made possible by tapping into the akasha. Interesting peculiarities are extant in all these evidences:

- they all are a posteriori ones – when a person experiencing some good or bad influence tries to find his aforegoing feelings and anticipations, and, as a rule, succeeds in it;
- no one journalize thoroughly his life from this point of view – in order for to check in future: how many premonitions were successful, and how many – not – which could facilitate calculation of reliable figures;
- we obtain refusal to organize reliable experiment which within the sound statistics could help in finding actual efficiency of such premonitions.

All phenomena considered in this section emphasize the outstanding importance of human psychology which together with socio-psychological egregorial constructs alone could be responsible for vast majority of situations, which, from the viewpoint of esotericism, should be attributed to fine-material, magic, or occult occurrences.

Conclusion

While being enthusiast of knowledge engineering approach to AI, it occurred to me, that we, trying to simulate human conscious and behavior, probably have rather vague understanding of what human intelligence really is, how does social life influence our mental sphere, how to overcome the immanent complexity barrier due which we inevitably bump into the concrete wall of controllability, or even, understandability when are trying to simulate really nontrivial behavior. My attention was attracted by esotericism with its very long history, quite different explanations of conscious – subconscious – superconscious, even its extravagant attitude to basic questions of philosophy. It was interesting to learn what was this exciting discipline about, and what the moral could be drawn from there for supporting progress in AI. Here are my findings.

1. According to evidences obtained by me from 2003, esotericism comprises neurosomatics, psychology, socio-psychology and rampageous fantasy frequently taking the form of mystic theories. No decisive indications of energy-information field, telepathy or clairvoyance were discovered. Of course, obtaining such evidences in future could change the attitude of nature-science specialists to this field.
2. The role of psychology, especially, socio-psychology in creation of human consciousness is tremendous. AI cannot be created through preprogramming. On this way we should expect not more than Mowgly being. Evolutionary approach towards AI should include, first of all, acknowledgement of SELF. And, of course, it is possible only in the frame of integral robot with the complete set of receptors, effectors and possibility to interact with the physical world and other beings, communities.
3. Materialistic approach to the problems of noosphere – is the sound foundation to the concepts of sign, information, knowledge, ideal object, intellectual subject, while the whole construct moulders to clay if we start our consideration from concepts, ideas, “word”.
4. The notion of belief was considered. It occurred that there are about 10 different kinds of beliefs. Beliefs themselves are indispensable part of human conscious activity.
5. The key notion of social activity – egregors was completely purified from any occult interference: fine-matter fields, “energies”, General Universe Intelligence. Their functioning can be completely specified from the positions of socio-psychology. At the same time, all humans live in egregorial jungles, and their real influence on genesis and behavior of humans is immeasurable.
6. Many “mysterious” effects, such as, magic, powerful spell of virtual communities, trolling, human fate and premonition could be naturally explained in terms of egregorial influences.

Acknowledgements. Research efforts in such mysterious sphere as esotericism, consciousness, psychological, clerical and psychic issues appeared to be an utmost entangled and uncertain serendipity. It was very encouraging that on this way one could meet such brilliant thinkers as Wladimir Korsuns'kyi [Korsuns'kyi, 2002] and Nick Fornit [Nan], to whom I express with pleasure my heartiest thanks and gratitude.

Bibliography

[Aldani] Lino Aldani, Onirofilm, Buonanotte Sofia, (nv) mag/giu 1963, *Futuro 1*, Futuro, Roma, http://litera-a.ahaha.ru/01/Aldany/aldany_onirofilm.htm

[Ashby] W. Ross Ashby, An Introduction to Cybernetics, Chapman & Hall, London, 1956, http://pespmc1.vub.ac.be/books/IntroCyb.pdf

[Augustine] St. (Blessed) Augustine of Hippo, http://en.wikiquote.org/wiki/Augustine_of_Hippo

[Bernstein] L.S. Bernstein, Egregor, Rosicrucian Archive, http://www.crcsite.org/egregor.htm

[Bible] The Gospel According to Saint John, The Project Gutenberg Etext of The King James Bible, http://www.gutenberg.org/etext/30

[Defamiliarization] Остранение, http://en.wikipedia.org/wiki/Defamiliarization

[Egregors, 1] ВикипедиЯ, http://ru.wikipedia.org/wiki/%D0%AD%D0%B3%D1%80%D0%B5%D0%B3%D0%BE%D1%80

[Egregors, 2] Определения «Аримойя», http://www.arimoya.spb.ru/Glossary/egregor.html

[Egregors, 3] http://wiki.traditio.ru/index.php/%D0%AD%D0%B3%D1%80%D0%B5%D0%B3%D0%BE%D1%80

[Egregors, 4] http://www.x-libri.ru/elib/klsku001/00000299.htm

[Korsuns'kyi, 2002] В.М.Корсунський, Засади сучасного наукового світогляду (Foundations of Contemporary Weltanschauung (Worldview)), Киев, 2002 г.

[Lozovskiy, 1990-1] В.С.Лозовский, Сетевые модели, разд. 1.3 в кн.: Искусственный интеллект, в 3-х кн., Кн. 2: Модели и методы. Справочник, п/р Д.А.Поспелова, М., "Радио и связь", 1990, стр. 28 - 49

[Lozovskiy, 1990-2] В.С.Лозовский, Инженерия знаний: понятия и компоненты, в сб. "Компьютерная революция и информатизация общества", Философское общество СССР, Секция "Методологические и социальные проблемы информатизации общества, Москва, 1990, стр. 112-129

[Lozovskiy, 1990-3] В.С..Лозовский, Правильным ли путем идем, товарищи?, II Всесоюзная конференция "Искусственный интеллект-90", 21-24.10.90., Круглые столы, САИИ, НС по проблеме "ИИ" АН СССР, ИТК АН БССР, Минск, 1990, стр. 38-42

[Lozovskiy, 1992] В.С.Лозовский, Есть ли будущее у инженерии знаний?, III Конференция по ИИ, КИИ-92, Сб. научных трудов, т. 1, Ассоциация ИИ, Тверь, 1992, стр. 106-110

[Lozovskiy, 1996] Vitaly S.Lozovsky, Semiotics of Net Models for Knowledge Representation, 12th European Conference on Artificial Intelligence ECAI'96, W30, Applied Semiotics, Budapest, 11-16 August, 1996, ECCAI, pp. 38-42

[Lozovskiy, 2001] V.S.Lozovskiy, Towards Parasemiotics of Loose Domains, Труды международной научно-практической конференции KDS-2001 «Знание – Диалог – Решение», том II, Северо-Западный государственный заочный технический университет, Санкт-Петербург, 17-22 июня 2001 года, ISBN-5-8114-0367-4, с. 425-432.

[Lozovskiy, 2003] V.Lozovskiy, Towards the semiotics of Noosphere, International Journal “Information Theories & Applications”, ISSN 1310-0513, Ed. in chief Krassimir Markov, 2003, Vol. 10, # 1, http://www.foibg.com, p. 29-36

[Lozovskiy, 2005] Vitaliy Lozovskiy, Approaching the Noosphere of Intangible - Esoterics from Materialistic Viewpoint, XI International Conference «Knowledge-Dialogue-Solution», June 20-24, Varna, Bulgaria, KDS-2005 Proceedings, Vol. 2, FOI-Commerce, Sofia, 2005, p. 657-668

[Nan] Nick Fornit, Теории, http://www.scorcher.ru/mist/theory.php , О мистике, http://www.scorcher.ru/mist/mist2.php , Мистические миры, http://www.scorcher.ru/mist/mist.php

[Trolling] Troll (internet), http://en.wikipedia.org/wiki/Troll_%28internet%29

Authors' Information

Vitaliy Lozovskiy – *Self-Employed, 35-21 Koroliova St., Odessa, 65113, Ukraine, e-mail: vitaaliy@gmail.com , URL: http://vitaliy.webhop.org/*

РАЗВИВАЮЩИЕСЯ СИСТЕМЫ

Александр Резник

***Аннотация**: Развивающаяся система определена как детерминированная система, способная к размножению. Рассмотрена эволюция популяций таких систем в реальном окружении и показана возможность их интеграции в сложные макроситемы. Поведение развивающейся системы представлено моделью стохастической динамической системы. Предложено использованием инфинитезимального оператора для описания развития в компактной аддитивной форме.*

***Ключевые слова**: развитие, система, информация, модель, генотип.*

Введение

Понятие развития можно относить к различным сущностям. Говоря о развитии растения, химической реакции или теории, мы имеем в виду различные объекты. В первом случае это материальный объект, во втором – процесс, а в последнем – понятие, существующее в сознании ученых. Во всех этих случаях речь идет о некоторых внутренних изменениях, при которых сущность объекта сохраняется. Поэтому подобные объекты можно рассматривать как своеобразные динамические системы, отличающиеся от обычных, определяемых в теории систем как отношения на множестве объектов [1], тем, что формально представить это первичное множество объектов невозможно. Изменения поведения системы могут быть связаны с адаптацией к неизвестным внешним факторам. Такие изменения могут быть представлены моделью целенаправленного поведения, содержащей заранее заданную цель и критерии ее достижения [2,3]. В этом случае для управления состоянием системы используются формальные правила, отвечающие этим критериям. Однако часто цель поведения, ни критерии ее достижения неизвестны. Смысл существования таких систем состоит в сохранении самих себя в реальных условиях окружения. Именно такие системы можно считать развивающимися [4].

Динамические системы и внешняя среда.

В общей теории систем динамическую систему определяют как многоместное отношение на множестве пар временных объектов $\{X_t, Y_t\}$: $X_t \in \Xi$, $Y_t \in \mathrm{H}$, $t \in T$. Здесь T - линейно упорядоченное множество моментов времени, X_t и Y_t - текущие значения входного и выходного объектов системы. Упорядоченность моментов времени позволяет представить это отношение последовательностью $\{X_t, Y_t, A_t\}$, где $A_t \in \Re$ - текущее состояние системы. Такое представление соответствует модели «черного ящика», в которой значения X_t и Y_t доступны для наблюдения, а состояние A_t является скрытым параметром системы. Поведение динамической системы описывают двумя соотношениями:

$$Y_t = F(X_t, A_t), \quad (1)$$

$$A_t = \Phi(A_{t-\tau}, X_{t-\tau}^{t}), \quad (2)$$

первое из которых называют уравнением вход/выход, а второе – уравнением состояний. Величина $X_{t-\tau}^{t} \in R^{n \times T}$ представляет реализацию n-мерного вектора X_t на интервале наблюдения $(t-\tau, t)$. Состояние $A_t \in \Re \subset R^{n \times \mathrm{T}}$ отражает предыдущее поведение системы. Мощность множества состояний $|\Re|$ характеризует степень сложности системы.

Выражения (1-2) определяют детерминированную динамическую систему. Считается что функции $F(.)$ и $\Phi(.)$ однозначно отображают элементы области определения, ($\Xi \times \Re$ и $\Re \times \Xi \times T$) в области значений (H и $\Re$, соответственно). Иногда допускается, что эти функции являются стохастическими, т.е. такое отображение не является однозначным и значение функции является случайной величиной. В этом случае содержание понятия система становится неопределенным.

Реально динамическая система существует в некотором окружении, соответствующем инверсной динамической системе, выход которой является входом данной системы, а вход - ее выходом. Предполагается что окружение намного сложнее данной системы, поэтому состояние инверсной системы $B_t \in \aleph$ при $|\aleph| >> |\Re|$ не зависит от состояния данной системы. Уравнения вход/выход окружающей среды можно представить как инверсию формулы (1)

$$X_t = F^*(Y_t, B_t). \tag{3}$$

Условием сосуществования системы и ее окружения является выполнение равенства:

$$Y_t = F[F^*(Y_t, B_t),\ A_t]. \tag{4}$$

Данное условие равнозначно требованию эквивалентности множеств $\aleph \equiv \Re$, что противоречит исходному предположению о независимости текущих состояний системы и ее окружения. Если отказаться от требования $\aleph \equiv \Re$, то выполнение условия сосуществования становится случайным событием. Это условие может выполняться в пределах конечного интервала θ для случайного значения реализации входа $X_{t-\theta}^{t}$, при котором поведение системы и ее окружения согласуются. Интервал θ, определяющий время жизни детерминированной системы в реальном окружении, является случайной величиной, зависящий от текущего значения реализации входа $X_{t-\theta}^{t}$.

Общая теория систем базируется на неявном предположении о том, что внешняя среда является открытой системой. Состояние такой системы не определено, поэтому считается, что внешняя среда не влияет на поведение детерминированной системы, заданной формулами (1-2). Во многих случаях это предположение оправданно, поскольку влияние окружения оказывается незначительным. Исключение, до последнего времени, составляли лишь явления микромира, для которых оказалось невозможным проводить наблюдения, не влияя на поведение наблюдаемых объектов. Чтобы обойти эту проблему, было введено соотношение неопределенности, позволившее вместо детерминированной модели системы использовать статистическую квантовую модель явлений микромира. Хотя попытки применить квантовую модель для более широкого класса систем (см., напр. [5]) особого успеха не имели, тем не менее статистический подход к описанию поведения систем получил развитие в теории адаптивных систем, стохастических автоматов, распознавании образов [2,3,6].

Фундаментальные и развивающиеся системы

Принятое в общей теории систем предположение об открытости окружения эквивалентно допущению, что внешняя среда заведомо удовлетворяет условиям (4). Это равнозначно наличию некой капсулы, изолирующей систему от внешнего мира, при условии, что окружение внутри капсулы представляет собой инверсную систему. Решениями уравнения (4) в этом случае являются собственные функции внутренней среды капсулы. Для выполнения условий согласованности функции F(.) и F*(.) должны отвечать этим решениям, и допускать разложения по собственным функциям внутри капсулы Системы, отвечающие этим требованиям, будем называть фундаментальными системами для внутреннего пространства капсулы. Из них могут быть построены любые системы, способные существовать в этом пространстве.

Если капсула охватывает все пространство, то фундаментальные системы воплощают общие законы. Например, долгоживущие элементарные частицы, отражающие фундаментальные законы Природы.

Существование, наряду с долгоживущими, также и короткоживущих элементарных частиц указывает на то, что одна и та же среда может содержать как фундаментальные системы, так и нефундаментальные системы, назовем их реальными, для которых условия согласованности могут нарушаться. Реальная система не является детерминированной, поскольку для продолжения существования после нарушения условия (4) данный экземпляр системы должен заменяться другим, начальное состояние которого заведомо удовлетворяет этому условию. Поведение такой системы представляет релаксацию - чередование спокойных (латентных) периодов скрытого накопления изменений и скачкообразных переходов в новое начальное состояние.

В общем случае скачкообразные изменения состояния детерминированной системы можно представить как действия некой суперсистемы, проверяющей выполнение условий согласования (4) и приводящей в действие механизм релаксации при их нарушении. Этот механизм может воздействовать как на данную систему, так и на ее локальное окружение. В первом случае создаются копии системы, отличающиеся начальным состоянием, и отбираются наиболее удачные экземпляры. Во втором случае предполагается, что скачок состояния выполняет локальное окружение данной системы, отделенное от внешнего мира капсулой. Иначе говоря, функции развивающейся выполняет не детерминированная система, а ее локальное окружение. Скачкообразное изменение состояния может происходить как перемещение границ капсулы или обмен элементами между внутренней и наружной средой капсулы. Это соответствует обмену веществ, наблюдаемому в живой природе.

Системы, способные преодолевать нарушение условий согласования (4) путем скачкообразного изменения своего состояния, будем называть развивающимися системами. Очевидно, это достаточно сложные, возможно, уникальные системы. До настоящего времени подобные долгоживущие системы встречаются только в живой природе. Безрезультатность предпринятых до сих пор попыток найти проявления жизни за пределами Земли лишь подтверждает уникальность таких систем.

Простейшей развивающейся системой, является популяция, члены которой (индивиды) представляют собой элементарные системы. Динамику изменения распределения численности популяции описывает стохастическая составляющая поведения развивающейся системы. Детерминированная составляющая поведения характеризует поведение членов популяции в пределах времени их жизни. Процесс эволюции начался, вероятно, с появления нефундаментальных систем, способных к релаксации, из которых образовались примитивные системы, способные к размножению. Рост популяций таких систем приводил к заполнению пространства вокруг каждой из них другими подобными ей системами и усилению их взаимного влияния внутри популяции. Имея одинаковый генотип, примитивные системы могли интерпретировать реакции соседей и соответственно изменять собственные реакции. В ходе эволюции простейшие развивающиеся системы приобретали способность координировать реакции членов своих популяций на воздействия внешней среды. В результате этого происходила интеграция популяций, их превращение в целостные макросистемы. Популяции примитивных систем образовывали примитивные макросистемы. В ходе эволюции из популяции примитивных макросистем возникали более сложные, что вело к формированию иерархии все более сложно организованных организмов, представляющих современную живую природу.

На вход каждого члена макросистемы действует композиция, состоящая из реакций других членов и воздействий внешней среды:

$$X_t^i = G^i V_{t_0}^t \oplus \Xi_t^i, \qquad (5)$$

где: $V_{t_0}^t = \{v_\theta^1, ..v_\theta^k, ..v_\theta^K\}_{t_0}^t$ - реализация совместной реакции K членов макросистемы;

G^i - оператор Грина;

Ξ_t^i - воздействие внешней среды на *i*-го члена макросистемы в момент времени *t.*

Составляющую вектора X_t^i можно представить в развернутом виде:

$$x_t^i(j) = \int_{t_0}^t \sum_{k=1}^K g(r^{i,j} - r^k, \tau) v_{t-\tau}^k d\tau + \xi_t^i(j), \quad (6)$$

где: $r^{i,j}, r^k$ - координаты *j* -го входа *i*-й и выхода *k*-го члена макросистемы;

$g(r^{i,j} - r^k, \tau)$ - передаточная функция окружающей среды;

$\xi_t^i(j)$ - составляющая воздействия внешней среды на *j*-й вход *i*- го члена макросистемы.

Соотношения между членами композиции (5) для различных членов макросистемы могут существенно отличаться. Эти различия минимальны в дисперсных макросистемах, состоящих из удаленных друг от друга элементарных систем. Для них преобладающим является влияние внешней среды. Примером дисперсной макросистемы может служить колония бактерий, для которой объединяющим фактором являются условия общей внешней среды. Размножение бактерий осуществляется путем деления индивидов, которые выжили на протяжении латентного периода. Деление обеспечивает передачу эстафеты жизни наследникам и почти не влияет на условия существования остальных бактерий колонии.

В консолидированных макросистемах, будем называть их организмами, состоящих из тесно расположенных членов популяции, поведение каждого из них больше зависит от реакций соседей, чем от воздействия внешней среды. При этом соотношение между составляющими входного воздействия зависит от позиции внутри организма. Для периферийных членов преобладающим является влияние внешней среды, тогда как для членов, находящихся внутри организма, это влияние может быть ничтожным. Внешние воздействия поступают внутрь организма в опосредствованной форме реакций пограничных элементарных систем. Для интерпретации этих реакций требовалось усложнение и функциональная специализация элементарных систем, образующих организм. Примеры такой специализации можно наблюдать в строении многоклеточных живых организмов. В процессе индивидуального развития организма клетки, находящиеся в различных условиях относительно внешнего окружения специализируются для выполнения различных функций. Например, клетки, непосредственно контактирующие с внешней средой, выполняют функции рецепторов и эффекторов. Функцией рецепторов является формирование внутреннего представления действующих извне стимулов для остальных клеток, специализирующихся на выполнении других функций живого организма. Эффекторы формируют общую реакцию организма. Специализация клеток происходит в ходе индивидуального развития организма, поэтому все клетки организма обладают одинаковым генотипом, заданным зародышевой клеткой.

Появление многоклеточных организмов знаменовало новый этап эволюционного процесса. В отличие от примитивных размножающихся систем, являющихся детерминированными, с конечным временем жизни, каждый организм представляет собой растущую популяцию, и его поведение меняется вместе с ростом популяции. Как и детерминированная система, организм отвечает модели «черного ящика», т.е. имеет вход и выход, доступные для наблюдения, и ненаблюдаемое внутреннее состояние. Однако, в отличие от детерминированной системы, состояние которой однозначно зависит от реализации на входе, состояние организма определяется совокупностью состояний многочисленных членов популяции.

Стохастические динамические системы

В предлагаемой модели стохастической динамической системе (СДС) поведение рассматривается как изменение закона распределения вероятностей для состояния членов популяции. Членами популяции (индивидами) могут быть детерминированные системы, либо более простые СДС. Примитивными будем считать СДС, индивидами которых являются детерминированные элементарные системы.

Поведение индивида определяет генотип ЭС – не зависящая от времени условная вероятность $\Delta(Y/X,A)$, где Y, X - значения выхода и входа, $A \in \Re$ - состояние ЭС. В СДС понятие состояния ассоциируется не с величиной A, как в детерминированной системе, а с текущим распределением вероятностей состояния индивидов $P_t(A)$. В примитивной СДС генотип представляет дельта-функция

$$\Delta(Y/X,A) = \delta\big(Y - F(X,A)\big),$$

что соответствует детерминированному уравнению вход/выход (1).

Стохастическим эквивалентом уравнений вход/выход и состояния являются [4,7]:

$$P_t(Y/X) = P_t(A)\Delta(Y/X,A), \tag{7}$$

$$P_t(A) = P_{t-\tau}(A)Q(A_{t-\tau}, A_t, X_{t-\tau}^t) \tag{8}$$

где: $Q(A_{t-\tau}, A_t, X_{t-\tau}^t)$ - стохастический оператор развития, описывающий марковский процесс перераспределения состояний индивидов при поступлении реализации $X_{t-\tau}^t$.

Легко показать, что СДС представляет обобщение детерминированной системы. Для этого достаточно представть уравнения (5,6) дельта-функциями:

$$\begin{aligned} \Delta(Y/X,A) &= \delta\big(Y - F(X,A)\big) \\ P_t(A) &= \delta\big(A_t - \Phi(A_{t-\tau}X_{t-\tau}^t)\big) \end{aligned}. \tag{9}$$

Очевидно, что полученные соотношения идентичны формулам (1, 2).

Если значения оператора развития не зависят от времени, то марковский процесс, описываемый уравнением (6) становится стационарным, и для него может существовать финальное распределение $P_\infty(A)$. Рассмотрим случай, когда оператор развития допускает представление

$$Q(A_{t-1}, A_t, X_{t-1}^t) = \alpha E + (1-\alpha)\Lambda, \tag{10}$$

где: *E* – единичный оператор $\Re$ x $\Re$, *α* – константа 0< *α<1*, Λ – стохастическая матрица:

$$\Lambda = \begin{bmatrix} \lambda_1 & \lambda_1 & \dots & \lambda_1 \\ \lambda_2 & \lambda_2 & \dots & \lambda_2 \\ \dots & \dots & \dots & \dots \\ \lambda_R & \lambda_R & \dots & \lambda_R \end{bmatrix}, \qquad \sum_{i=1}^{\Re} \lambda_i = 1. \tag{11}$$

В этом случае формула (7) приобретает вид:

$$P_{t_T}(A) = \alpha^T P_{t_0}(A) + (1-\alpha^T)\lambda = \xrightarrow[t_T \to \infty]{} \lambda, \tag{12}$$

где λ - вектор-столбец матрицы Λ.

Полученное выражение описывает стохастическую модель обучаемости [8], применяемую при изучении динамики выработки условного рефлекса у подопытных животных.

Уравнения (7-8) отражают точку зрения внешнего наблюдателя, рассматривающего реализацию $X_{t-\tau}^{t}$ как причину изменения распределения $P_t(A)$. С точки зрения самой СДС окружающая среда является внешней системой, реагирующей на реакции СДС. Поведение внешней среды описывается стохастическими уравнениями

$$P_t(X/Y) = P_t(B)\nabla(X/Y,B); \tag{13}$$

$$P_t(B) = P_{t-\tau}(B)R(B_{t-\tau}, B_t, Y_{t-\tau}^{t}), \tag{14}$$

где: $\nabla(X/Y,B)$ - генотип внешней среды;

$P_t(B)$, - текущее распределение состояний индивидов внешней среды;

$R(B_{t-\tau}, B_t, Y_{t-\tau}^{t})$ - оператор развития для внешней среды.

Уравнения (13-14) представляют стохастическую модель, которую можно выбрать так, чтобы множества состояний индивидов СДС и ее окружения совпадали $A \equiv B \in \Re$. Такую модель будем называть собственной моделью внешней среды. Для собственной модели можно сформулировать соотношения, аналогичные условиям согласованности (4) для детерминированной системы:

$$Q(A_{t-\tau}, A_t, X_{t-\tau}^{t}) = R(B_{t-\tau}, B_t, Y_{t-\tau}^{t}). \tag{15}$$

Смысл данного соотношения состоит в том, что различие в поведении СДС и ее окружения выражается как несоответствие собственной модели реальной внешней среде. Возникающее вследствие этого расхождение между прогнозируемым на основе реализации $Y_{t-\tau}^{t}$, и реальным значениями входа СДС может служить сигналом для соответствующего изменения состава популяции.

Инфинитезимальный оператор

Полагая, что распределение вероятностей состояний $P_t(A)$ является непрерывной функцией времени, найдем производную:

$$\frac{\partial}{\partial t}\left[P_t(A)\right] = P_{t-\tau}(A)\frac{\partial}{\partial t}Q(A_{t-\tau}, A_t, X_{t-\tau}^{t}). \tag{16}$$

Если на интервале времени $(t-\tau, t)$ эта производная существует, то можно найти ее предел:

$$\frac{\partial}{\partial t}\left[P_t(A)\right] \xrightarrow[\tau\to 0]{} P_t(A)H(A_t, X_t) \tag{17}$$

Величина $H(A_t, X_t)$ является инфинитезимальным оператором СДС, представляющим скорость изменения вероятности пребывания индивида системы в данном состоянии при текущем значении входа системы. Отрицательное значение $H(A_t, X_t)$ указывает на рост вероятности гибели индивида, а позитивное - на необходимость увеличения числа индивидов в состоянии A_t, т.е. их размножения.

Если $P_t(A) \neq 0$, (это справедливо, если в популяции представлены все возможные состояния $A_t \in \Re$), дифференциальное уравнение (14) имеет решение:

$$P_t(A) = P_{t_0}(A)\exp\left[\int_{t_0}^{t} H(A_\theta, X_\theta)\partial\theta\right]. \tag{18}$$

Интеграл в показателе экспоненты описывает процесс накопления изменений поведения СДС в ходе ее эволюции. Аддитивный характер изменений и их зависимость от текущих значений входа СДС X_t,

позволяет интерпретировать такие изменения, как накопление информации о поведении окружающей среды. Действительно, количество информации, получаемой СДС, при поступлении на ее вход значения X_t определяется выражением:

$$h(A_t, X_t) = \xrightarrow[\tau \to 0]{} \sum_{\Re} \{P_t(A) \log[P_t(A)] - P_{t-\tau}(A) Q(A_{t-\tau}, A_t, X_{t-\tau}^t) \log[P_{t-\tau}(A) Q(A_{t-\tau}, A_t, X_{t-\tau}^t)]\} = \\ = -\sum_{\Re} P(A_t) H(A_t, X_t) \quad (19)$$

Соотношения, аналогичные (16-18) можно получить и для внешней среды:

$$\frac{\partial}{\partial t}[P_t(B)] = P_t(B) G(B_t, Y_t) \\ P_t(B) = P_{t_0}(B) \exp\left[\int_{t_0}^{t} G(B_\theta, Y_\theta) \partial\theta\right] \quad (20)$$

Здесь величина оператора $G(B_t, Y_t)$ характеризует влияние реакции индивида популяции на изменение распределения состояний внешней среды. Для собственной модели внешней среды значения инфинитезимальных операторов СДС и ее окружения совпадают:

$$H(A_t, X_t) = G(A_t, Y_t) . \quad (21)$$

Такая связь между текущими значениями инфинитезимального оператора может использоваться для внешнего управления поведением СДС. Для этого значения $G(A_t, Y_t)$ должны поступать в нее извне, например, в форме дополнительной компоненты входа X_t. Такое управление соответствует обучению с подкреплением.

Развитие консолидированных макросистем

Областью приложения теории СДС являются организмы - консолидированные макросистемы, которые, в отличие от простейших и дисперсных развивающихся систем, являющихся однородными популяциями, представляют экземпляр единственной популяции, жизнь которой начинается с зародышевой ЭС и заканчивается при исчерпании возможности ее дальнейшего роста. Вначале развития организма увеличивается численность популяции и происходит специализация ее членов в соответствии с их расположением внутри организма. В этой фазе развития организм требует особой защиты от окружения, которое может влиять на образование и специализацию новых членов популяции. Поэтому начальный этап развития происходит в искусственном окружении. На высоких стадиях эволюции таким окружением является утроба матери или оболочка яйца, а после появления на свет – сообщество, т.е. множество окружающих организмов, поддерживающих развитие новых членов до достижения ими стадии зрелости, когда они смогут производить зародышевые элементарные системы, воспроизводящие генотип данного организма.

Коллективное существование макросистем в форме сообществ, состоящих из организмов, находящихся в различных стадиях развития, требовало развития средств коммуникации между членами сообщества, создавало условия конкуренции, в которых преимущество получали члены сообщества, способные лучше и быстрее реагировать на окружающую обстановку. Это привело к появлению нервной системы, что стало поворотным пунктом эволюции.

До появления нервной системы каждый организм оставался детерминированной системой, поэтому изменения его поведения могло происходить лишь путем размножения- гибели, т.е. естественного отбора экземпляров с необходимыми свойствами. С появлением нервной системы организмы приобрели способность моделировать различные варианты поведения и проверять их, пользуясь моделью внешнего

мира, запечатленной в их памяти. Имея нервную систему, каждый организм мог оперативно изменять свое поведение, реагируя на изменения окружения. Подобные изменения поведения при отсутствии нервной системы, потребовались бы многих поколений развития.

Появление нервной системы не только ускорило эволюционный процесс, но и сделало его намного более экономным, поскольку отпала необходимость в громадном увеличении численности популяций. Соответственно сократилась нагрузка на окружающую среду, что высвободило ее ресурсы для ускорения эволюции. Высокоразвитые организмы получили возможность активно использовать окружающую среду для улучшения условий инкапсуляции. Преобразование локального окружения способствовало появлению сознательного труда, формированию интеллекта. Дальнейшее все ускоряющееся развитие организмов привело к образованию цивилизации, формированию индустриального и постиндустриального общества.

В эпоху цивилизации взаимодействие организмов с окружением можно рассматривать как симбиоз, в котором объекты окружения приобретают свойства самостоятельных развивающихся систем, а сообщество - среды, способствующую их существованию. Образуется мир виртуальной реальности, наполненный виртуальными развивающимися системами, которые готовы или вынуждены поддерживать члены сообщества. Виртуальными системами могут быть такие объекты как: технологии, виды деятельности, идеи и верования, социальные движения, и т.п. Вовлекая сообщество в свою поддержку, они приобретают свойства самостоятельных развивающихся систем, существующих в благоприятном окружении создаваемом вниманием и усилиями его членов. Убедительными примерами виртуальных систем может служить развитие технологий, вооружений, машин, компьютеров и т.п. Разработчики и производители подобных объектов образуют, по сути, генотип сложного развивающегося организма. Занятые совершенствованием своего детища, созданием новых, более эффективных его поколений, они образуют ядро большой системы, включающей также подсистемы кооперации, торговли и рекламы, которые заняты преобразованием сообщества в среду, благоприятную для процветания этой системы. Подобной схеме отвечает развитие любой крупной системы от промышленной корпорации или научной лаборатории до политического движения или партии. В последнем случае генотип системы определяется амбициями лидеров, скрытыми за оболочкой из привлекательных для сообщества лозунгов.

Если прослеживать развитие исторических событий, технологических проектов, научных теорий или политических интриг, то везде можно обнаружить существование популяций, индивидами которых являются участники или последователи. В развитии таких популяций существуют периоды роста, зрелости, деградации или перерождения, которые легко прослеживаются на графиках показателей развития, которые выглядят практически одинаково, идет ли речь о популяции динозавров [9], доходах корпорации [10], количестве публикаций или числе сторонников [11,12].

Библиография

1. М. Месарович, Я. Такахара Общая теория систем. Математические основы. – М. Мир. 1978. – 311с.
2. Цыпкин Я. З. Адаптация и обучение в автоматических системах. – М. Наука. 1968. –399с.
3. Срагович В.Г. Теория адаптивных систем. - М. Наука. !976. – 317с.
4. Різник О.М. Загальна модель розвитку. // Математичні машини і системи. -2005. -№ 1. –С. 84-98.
5. Хелстром К. Квантовая теория проверки гипотез и оценивания. – М. Мир. 1979. – 344с.
6. Vapnik V.N. Statistical learning theory. - John Willey & sons inc. 1998, - 736p.
7. А.М. Резник Многорядные динамические перцептроны / в кн. Перцептрон- система распознавания образов, ред. А.Г. Ивахненко. –Киев. Наукова Думка. 1975.- с. 243-292.
8. Буш Р., Мостеллер Ф. Стохастические модели обучаемости. – М. ГИФМЛ. 1962. – 483с.
9. Грант В. Эволюция организмов. –М. Мир. 1980.-407с.

10. Янч Э. Прогнозирование научно-технического прогресса.
11. Кун Т.. Структура научных революций. - М. Прогресс. 1975.- 246с.
12. Михайлов А.И., Черный А.И., Гиляревский Р.С. Научные коммуникации и информатика. – М. Наука. 1976 435с.

Информация об авторе

Александр Михайлович Резник *– Институт математических машин и систем НАН Украины, зав. отделом нейротехнологий, Киев просп. Академика Глушкова 42, e-mail neuro@immsp.kiev.ua*

О ПРИРОДЕ ИНТЕЛЛЕКТА

Александр Резник

Аннотация: *Предложен новый подход к объяснению механизмов нервной активности и свойств интеллекта, основанный на модели эволюции как способа существования развивающихся систем в произвольном окружении. Развивающаяся система определена как детерминированная система, обладающая свойствами размножения и обмена веществ. Рассмотрено, как из таких систем образуются макросистемы, происходит специализация составляющих и появляется нервная система, выполняющая функции моделирования поведения макросистемы. Показано, что импульсные потоки нервной активности воспроизводят процесс естественного отбора стереотипов поведения.*

Ключевые слова: *интеллект, развитие, система, нейрон, модель.*

Введение

Понятие интеллекта ассоциируется со способностью ориентироваться в незнакомой ситуации, находить нетривиальные решения сложных задач или умением делать правильный выбор из множества альтернатив. Моделирование этих способностей и построение искусственных интеллектуальных систем является актуальной задачей современной науки. Более полувека назад сформировались два основных подхода к ее решению. Один, связанный с моделированием мышления и разумного поведения, изучаемого экспериментальной психологией, получил название ”искусственный интеллект” [1]. Второй, основанный на моделировании структур и функций нервной системы, исследуемой нейрофизиологами, называют ”искусственные нейронные сети” [2]. Общей целью этих, диаметрально противоположных направлений, является создание интеллектуальных прикладных систем, воспроизводящих функции мозга. Прикладная направленность этих направлений отличает их от комплексного подхода к изучению функций мозга, принятого в биологической кибернетике, дающей достаточно полное общее представление о строении, развитии и функциях мозга. [3,4,5]. Однако, к сожалению, отвечая на вопрос, как устроен и работает мозг, эта наука пока оставляет без ответа вопрос, почему он устроен и функционирует так, а не иначе. Ответ на этот вопрос мог бы нас приблизить к пониманию принципов организации хранения и обработки данных в мозгу, которые определяют саму природу интеллекта.

Мы предлагаем иной подход к пониманию природы интеллекта, основанный на моделировании процесса его образования и развития в ходе эволюции. Рассматривается модель системы, обладающей двумя базовыми свойствами: репликации (размножения с сохранением генотипа) и обмена веществ. Используя подход общей теории динамических систем, мы показываем, как такая система может сосуществовать со своим окружением, образовывать популяции, представляющие простейшие развивающиеся системы. В

процессе эволюции они могут образовывать макросистемы - прототипы многоклеточных организмов. Дальнейшая эволюция протекает как взаимное приспособление макросистем и их локального окружения, что ведет к появлению высших организмов, нервной системы и развитию интеллекта. Организация нервной системы образует специфичную среду, в которой происходит эволюция потока нервной активности, управляющего поведением организма. Параметры среды генетически запрограммированы и частично модифицируются при обучении.

Используемая в работе статистическая модель развивающейся системы представлена в [5].

Детерминированные и развивающиеся системы

Общая теория систем [6] для представления динамической системы использует модель “черного ящика”, в которой доступны для наблюдения вход и выход, а состояние является скрытым внутренним параметром системы. Система, значение выхода которой однозначно определяется текущими значениями ее состояния и входа, является детерминированной. Такое определение динамической системы не учитывает, что любая реальная система существует в окружении, являющемся инверсной динамической системой, выход которой отвечает входу данной системы, а вход - с ее выходу. Сосуществование этих систем требует полного согласования текущих значений их входов и выходов, что возможно лишь в случае их эквивалентности. В общем случае совпадение значений входа системы и реакции окружения является случайным событием, которое может продолжаться в течение конечного времени, определяющего продолжительность жизни детерминированной системы в данном окружении. Средняя продолжительность жизни падает с увеличением различия в сложности системы и ее окружения.

Общая теория систем исходит из предположения, что внешняя среда является открытой системой, что не мешает существованию любой детерминированной системы. Во многих практических случаях это предположение оправданно, однако в таких областях как явления микромира, экология, экономика и др. исключение влияния внешней среды недопустимо. Применительно к явлениям микромира введено понятие неопределенности, позволившее заменить детерминистский подход теории систем статистической квантовой моделью наблюдаемых явлений. Статистическая модель поведения используется также в теории стохастических автоматов, адаптивных систем, распознавания образов [7].

Принципиально иной подход к решению проблемы существования детерминированной системы в произвольном окружении, основан на понятии развивающейся системы, способной преодолевать свои расхождения с поведением окружения [8]. Этот подход основывается на утверждении, что величина расхождений между значениями входа детерминированной системы и реакции окружения может не выходить за конечные пределы лишь в течение конечного интервала времени. Для продолжения существования системы за пределами этого интервала необходима замена данного экземпляра системы новым экземпляром, начальное состояние которого заведомо отвечает текущему состоянию окружения. Такая замена экземпляров напоминает релаксацию - чередование спокойных (латентных) периодов скрытого накопления изменений со скачкообразными переходами в новое начальное состояние. Скачкообразные изменения состояний можно рассматривать, как действие некой суперсистемы, проверяющей согласованность поведения системы и окружения и приводящей в действие механизма релаксации при выявлении рассогласования. Механизм релаксации может воздействовать как на данную систему, так и на ее окружение. В первом случае создаются копии данной системы, с различными начальными состояниями, из которых отбирается экземпляр, наиболее отвечающий условиям окружения. Во втором случае состояние изменяет локальное окружение, которое можно представить как часть внешней среды, заключенной в некоторую капсулу, изолирующую от остального мира. Скачкообразные изменения свойств локального окружения могут происходить как перемещения границ капсулы, или как

обмен элементами с внешней средой. Такой способ согласования поведения детерминированной системы и ее окружения будем называть инкапсуляцией. Действие механизма инкапсуляции соответствует обмену веществ в живой природе.

Независимо от способа реализации механизма релаксации, его эффективность зависит от того, насколько удачным оказывается выбор следующего состояния. Такой выбор осуществляется путем естественного отбора. Для этого детерминированная система должна обладать способностью к воспроизводству, т.е. размножению. Из множества создаваемых при размножении потомков выживают лишь те экземпляры, которые наиболее соответствуют текущему состоянию окружения. Естественный отбор позволяет выявлять и сохранять любые незначительные отличия между родительскими и дочерними экземплярами системы, если они оказываются полезными для выживания. Очевидно, что подобными свойствами могут обладать лишь достаточно сложные, возможно, уникальные системы. Пока развивающиеся системы мы наблюдаем только в живой природе. Безрезультатность предпринятых до сих пор попыток найти проявления жизни за пределами Земли лишь подтверждает их уникальность.

Стохастические динамические системы

Последовательность релаксаций, происходящих в случайные моменты времени, характеризует стохастическую компоненту поведения развивающейся системы. Детерминированная составляющая относится к ее поведению в пределах латентных периодов. Наличие детерминированной и стохастической и составляющих поведения допускает различные их сочетания, конечные реализации которых могут совпадать. Поэтому развивающиеся системы могут приспосабливаться к изменениям условий окружения, меняя лишь стохастическую составляющую поведения и сохраняя без изменений его детерминированную компоненту.

Простейшей развивающей системой, является популяция, члены которой (индивиды) представляют собой элементарные системы, условно назовем их клетками. Динамику изменения распределения численности популяции описывает стохастическая составляющая поведения. Детерминированная составляющая характеризует поведение членов популяции в пределах времени их жизни. В процессе эволюции популяции клеток могут сливаться, образуя многоклеточные структуры. Поведения таких структур описывает модель стохастической динамической системы (СДС). Основным понятием этой модели является генотип, определяющий не зависящее от времени условное распределение вероятностей реакции клетки при заданных значениях ее состояния и входа. Состояние СДС представляет текущее значение распределения вероятностей состояний клеток. Примитивной будем считать СДС, клетками которой являются детерминированные системы, имеющие конечное время жизни. Описание математической модели СДС приведено в нашей работе [5].

В процессе развития детерминированная и/или стохастическая составляющая поведения претерпевают изменения. Изменения детерминированной составляющей меняют поведение клеток, т.е. генотип СДС. Изменения стохастической составляющей связаны с механизмом инкапсуляции и затрагивают, в основном параметры ближайшего окружения клеток СДС. Совокупность значений этих параметров образует внегенетическую память развивающейся системы.

Процесс эволюции начался, вероятно, с выделения из множества короткоживущих систем, обладавших свойством релаксации, элементарных систем, способных к размножению. Это были простейшие развивающиеся системы, образованные примитивными СДС. Рост популяций сопровождался заполнением окружающего пространства каждой клетки другими подобными ей клетками и усилением их взаимного влияния внутри популяции. Имея одинаковый генотип, клетки могли интерпретировать реакции соседей и соответственно изменять собственные реакции. Как результат, в процессе эволюции

примитивные СДС приобрели способность координировать реакции членов своей популяции на воздействия окружения. Это способствовало интеграции популяций, превращению их в целостные стохастические динамические макросистемы. При интеграции примитивных СДС образовывались примитивные макросистемы. Популяции таких макросистем в процессе эволюции могли образовывать высокоорганизованные макросистемы – прототипы живых организмов.

На вход каждой клетки, входящей в состав макросистемы, помимо воздействий внешней среды поступают реакций других клеток. Соотношения между ними для различных клеток могут существенно отличаться. Различия минимальны в дисперсных макросистемах, члены которых удалены друг от друга. Примером могут служить колонии бактерий, для которых взаимное влияние членов невелико, а объединяющим фактором являются общие условия внешней среды. Противоположность дисперсным составляют тесно сплоченные консолидированные макросистемы, будем их называть организмами, в которых поведение большинства членов зависит не столько от внешней среды, сколько от реакций их соседей внутри популяции. Прямое воздействие окружающей среды испытывают только наружные клетки. Клетки, расположенные внутри получают информацию о внешнем окружении опосредствовано, в форме реакций наружных клеток. В процессе эволюции такое различие привело к функциональной специализации клеток в соответствии с их расположением внутри организма. Углубление специализации происходило, в основном, путем инкапсуляции, но могло затрагивать и детерминированную составляющую, т.е. генотип СДС. При изменении генотипа клетки становились непригодными для воспроизводства СДС, поэтому в ходе эволюции у высокоорганизованных организмов появились специализированные репродуктивные органы, производящие зародышевые клетки воспроизводящие их генотип.

Развитие консолидированных макросистем

Каждый организм представляет популяцию, жизнь которой начинается с появления соответствующей зародышевой клетки. Время жизни члена популяции может быть намного короче, чем у организма, поэтому состав популяции может многократно обновляться. Развитие организма начинается с увеличения численности популяции и углубления специализации ее членов. Росту специализации способствует то, что каждое поколение членов популяции развивается в среде, созданной предыдущим поколением. Происходит как бы наслоение клеток вокруг ранее образовавшегося ядра, соответственно пути эволюции данного организма. Этот процесс можно наблюдать на примере эмбрионального развития млекопитающих. Известно, что на начальном этапе развития эмбрионов у них появляются жабры, характерные для гораздо более ранней стадии эволюции. Образовавшие жабры впоследствии исчезают.

Развитие зародышевой клетки сильно зависит от условий окружения, влияющих на образование и специализацию новых членов популяции. Поэтому формирование организма может происходить лишь в условиях инкапсуляции. В живой природе функцию капсулы у растений выполняют семена, у пресмыкающихся и птиц – яйца, в животном мире - утроба матери. После появления на свет функцию капсулы выполняет сообщество организмов, поддерживающее развитие новых членов до достижения ими стадии зрелости, когда последние сами смогут производить зародышевые клетки.

Сообщество организмов образует своего рода супермакросистему, в которой функции членов достаточно четко распределены. Старшие поколения формируют среду, способствующую воспроизводству и передаче опыта выживания следующим поколениям. В живой природе примерами сообществ, отвечающих различным стадиям эволюции, могут служить кораллы, колонии насекомых, стада животных, человеческое общество.

На ранних стадиях эволюции специализация клеток организмов носила внегенетический характер, что подтверждается вегетативной формой размножения у примитивных многоклеточных живых организмов.

Генетическая форма специализации стала заметной на более поздних этапах эволюции, когда появились более сложные организмы, обладающие специализированными репродуктивными органами. В эпоху грибов и папоротников их функцию выполняли споры. Позднее, с появлением половой системы размножения, зародышевае клетки разделились на яйцеклетки и сперматозоиды, формируемые различными особями. При двуполой системе размножения появлению каждого нового поколения предшествовала жесткая проверка идентичности кода родителей. Это не только гарантировало сохранение генотипа при сбоях репродуктивных органов, но и обеспечивало сохранение приобретаемых в ходе индивидуального развития небольших изменений генотипа, полезность которых подтверждалась индивидуальным опытом обоих родителей.

Нервная система

В сообществах, содержащих большое число различных организмов, неизбежно возникала конкуренция между их членами за доступ к общим ресурсам или более выгодное расположение внутри сообщества. Преимущества получали те из них, кто был способен быстрее оценивать ситуацию и оперативно на нее реагировать. Это привело к образованию в составе более развитых организмов особых клеток для управления поведением других клеток организма. Это были нервные клетки (нейроны), принципиальное отличие которых состояло в использовании электрохимических процессов формирования импульсных реакций, что позволяло быстро передавать реакции клетки на большие расстояния. Нейрон имеет разветвленную мембрану, усеянную синапсами, принимающими возбуждения от других нейронов. Поведение нейрона напоминает релаксацию: если суммарное возбуждение, поступающее на синапсы в течение короткого времени, превышает некоторый порог, нейрон переходит в возбужденное состояние и генерирует электрический импульс (спайк), который через разветвления его выходного волокна (аксона) передается остальным клеткам нервной системы. Нейрон способен различать определенные сочетания возбуждений, характер которых зависит от проводимости его синапсов. Проводимости (веса) синапсов устанавливаются в процессе обучения. Они отражают ассоциативные связи потоков спайков на входах и выходах каждого нейрона [9,10].

Разветвленная сеть связей каждого нейрона, охватывающая до нескольких тысяч нервных клеток, образует специфическую среду, в которой распространяется поток спайков, инициируемый рецепторами, воспринимающими внешние воздействия. Структура нервной сети формируется в ходе эволюции данного вида организмов, а распределение и вес межнейронных связей устанавливаются в процессе индивидуального развития, отражая индивидуальный опыт данного организма. При прохождении потока спайков через нейронную сеть происходит выделение запомненных ранее ассоциаций и восстановление соответствующих реакций сети.

На высоких уровнях эволюции происходит разделение нервной системы на центральную (ЦНС) и периферийную. Периферийная система выполняет исполнительные функции и реализует стереотипы поведения организма, закрепленные на генетическом уровне. Функция ЦНС состоит в интерпретации потоков спайков, поступающих от рецепторов и подготовке реакций на них. При прохождении через ЦНС поток спайков от рецепторов претерпевает преобразования, подобные процессу естественного отбора, описываемому моделью стохастической динамической системы. Преимущества получают те последовательности спайков, которые напоминают запомненные ранее. В периферийной системе такие последовательности преобразуются в соответствующие реакции организма.

Анализируя общую схему нервной активности можно составить определенное представление о динамических процессах, связанных с мышлением. Естественно связывать явление мышления с функцией неокортекса - новой коры головного мозга, появившейся на самых последних этапах

эволюционного процесса. Ткань неокортекса представляет складчатую структуру, содержащую более десятка слоев нейронов с сильно развитыми латеральными (внутрислойными) связями, число которых в сотни раз превосходит количество связей между слоями [11]. Более 90% связей нейрона двусторонние, поэтому каждый нейрон является членом группы, включающей до тысячи нейронов. Организация связей между нейронами в такой группе напоминает сеть Хопфилда, обладающую свойствами ассоциативной памяти [12]. Такая группа, называемая хопфилдовским ансамблем имеет устойчивые состояния (аттракторы), число которых составляет порядка 10% от количества нейронов в составе группы. Поступающие извне потоки спайков возбуждают резонансную активность, соответствующую аттракторам хопфилдовского ансамбля. Появление такой активности соответствует ассоциативному вспоминанию запомненного ранее возбуждения данного ансамбля.

Слои нейронов коры головного мозга представляют множество перекрывающихся хопфилдовских ансамблей, каждый из которых имеет собственный набор аттракторов, отражающий структуру запомненных потоков спайков. В потоке спайков, пересекающем несколько нейронных слоев, формируется последовательность, состоящая из аттракторов, представляющих фрагменты образов, ранее запомненных данным организмом. Потоки спайков, циркулирующие в глубинных слоях неокортекса, постоянно извлекают эти образы и ассоциации, воспроизводя фрагменты индивидуального прошлого данного организма. Анализ таких фрагментов, воссоздание из них картин прошлого и формирование новых образов собственно и составляет содержание мышления. Процесс мышления можно рассматривать как осознанное, т.е. достигшее исполнительных органов отражение потоков спайков, циркулирующих в глубинных слоях нейронов коры головного мозга. Большая часть нервной активности происходит на подсознательном уровне, проявляясь в сновидениях, галлюцинациях и внезапных озарениях[13].

Модель хопфилдовских ансамблей объясняет природу долговременной памяти, позволяющей сохранять яркие образы прошлого на протяжении жизни. Она связана с низкочастотной (менее 1гц.) спонтанной активностью нейронов, которая систематически регенерирует аттракторы хопфилдовских ансамблей коры мозга. При возбуждении, хопфилдовского ансамбля случайным потоком спайков последний переходит в ближайший к этому возбуждению аттрактор, представляющий некоторый фрагмент запомненного ранее возбуждения. Благодаря ассоциативным свойствам сети Хопфилда ее аттракторы сохраняются даже при разрушении части связей или нейронов. (Число удаленных нейронов не может превышать аттракторный радиус сети, равный половине отношения числа нейронов к числу аттракторов [14]). При небольших разрушениях сети аттракторы, возбужденные путем спонтанной активации хопфилдовских ансамблей, могут запоминаться повторно в тех же или в соседних хопфилдовских ансамблях. Таким образом, спонтанная активация хопфилдовских ансамблей, позволяет систематически обновлять содержимое памяти и сохранять ранее запомненные образы достаточно долго, несмотря на деградацию связей и отмирание отдельных нервных клеток мозга. Можно предполагать, что такой процесс регенерации памяти происходит, в основном, во время сна [15].

Природа интеллекта

Появление нервной системы существенно изменило ход эволюции. До появления нервной клетки каждый организм оставался детерминированной системой, поэтому изменения поведения могло происходить лишь путем роста популяции и естественного отбора экземпляров с необходимыми свойствами. С появлением нервной системы каждый организм приобрел способность моделировать различные варианты своего поведения и проверять их, пользуясь моделью внешнего мира, запечатленной в памяти. Благодаря этому совершая каждый поступок организм предварительно в сознании проверяет множество

вариантов поведения. При отсутствии нервной системы, такая проверка потребовала бы нескольких поколений развития.

Образование нервной системы не только ускорило эволюционный процесс, но и сделало его намного более экономным. Отпала необходимость в громадном увеличении численности популяций, для проверки все возрастающего числа возможных направлений релаксации. Соответственно сократилась нагрузка на окружающую среду, что высвободило ее ресурсы для ускорения процесса эволюции. Сообщества высокоразвитых КМС получили возможность активно использовать окружающую среду для улучшения условий своего существования, в результате чего преобразование окружающей среды со временем стало основным смыслом развития КМС.

Активная эксплуатация окружения началась еще на заре эволюции, о чем свидетельствуют ракушечные отложения мелового периода. Раковины, а позднее искусственные жилища типа нор и гнезд, построенные из элементов внешнего окружения, можно рассматривать как простейшие объекты, воплощавшие в себе полезный опыт, приобретенный в ходе эволюции их создателей. Появление таких искусственных объектов знаменовало начало развития средств внешней коммуникации, несущих функцию внегенетической памяти обеспечивающей как обмен опытом между членами сообщества, так и его передачу потомкам.

Использование внешних объектов для коммуникации требовало соответствующего приспособления естественных реакций организма. Результатом такого приспособления стало образование специальных органов, обеспечивающих визуальную, звуковую, тактильную и обонятельную коммуникацию. Примерами наипростейших средств коммуникации можно считать окраску или запах особей, стрекот цикад, танцы пчел, следы феромона у муравьев [16]. Наивысшими достижениями Природы в этой области можно считать членораздельную речь, язык свистов сильбо, используемый горцами, полигармонические сигналы речи дельфинов, позволяющие им поддерживать связь при движении в слоисто-неоднородной водной среде [17].

Дальнейшим развитием внешней коммуникации стало использованием элементов внешней среды как инструментов, продолжающих и/или дополняющих органы самих членов сообщества. Вначале это были случайно взятые предметы (палки, камни), но со временем такие предметы стали создаваться специально. Их появление стало возможным благодаря развитию способности к абстрактному мышлению, позволявшему оперировать с мысленными образами реальных объектов и событий. В этот же период началось развитие членораздельной речи, позволившее улучшить обмен информацией между членами сообщества, планировать совместные действия, устранять возможные конфликтные ситуации. Дальнейшее развитие средств коммуникации, привело к появлению материальной культуры, как носителя внегенетической памяти сообщества, и материального производства как основного занятия его членов.

Все это сопровождалось возрастанием роли интеллектуальной деятельности, которая со временем стала доминировать в разделении общественного труда. Превращение интеллекта в производительную силу сообщества стало новым этапом эволюционного процесса, в котором появился новый участник – продукты умственной деятельности, существующие в сознании членов сообщества. Часть этих продуктов достигала стадии воплощения в материальных объектах, которые затем, как носители интеллектуальных ценностей, начинали жить собственной жизнью. Остальные оставались в сознании членов сообщества, образующем среду для рождения, развития и гибели новых интеллектуальных продуктов. История науки и научных коммуникаций дают достаточное количество примеров зарождения и развития и угасания научных парадигм [18,19], дающих основания проводить параллели с процессами эволюции в живой природе.

Рассматривая общественное сознание как специфическую среду для развития интеллектуальных продуктов, нельзя не учитывать, что эта среда состоит из множества сознаний членов сообщества, которые собственно и производят интеллектуальные продукты, поступающие затем на суд сообщества. Дальнейшая судьба созданных продуктов зависит от их восприятия другими членами сообщества. Новые, непривычные предложения независимо от их ценности могут быть отвергнуты из-за непонимания. При наличии достаточного уровня восприятия, идея, выдвинутая членом сообщества, получает некоторое число сторонников, которые могут передавать ее другим. Идея превращается в популяцию своих копий, которые, распространяясь дальше, могут дополняться мыслями других людей. Происходит эволюция идей, в ходе которой эволюционируют не только сами идеи, но и среда, в которой они циркулируют. Таким образом, интеллектуальная деятельность представляет собой самоподдерживающийся виртуальный слепок эволюции, приведшей к появлению высокоорганизованных систем, способных мыслить.

Сходство с процессом эволюции напрашивается и при анализе того, что происходит в сознании отдельного человека при решении сложных задач. Выдвигаемые гипотезы и варианты решений, эвристические догадки – все это многократно в различных комбинациях проверяется, пока не выкристаллизуется приемлемое решение. Огромная работа при этом выполняется на подсознательном уровне и поэтому не замечается. Ее выполняют потоки спайковой активности, циркулирующие в нейронных слоях коры мозга. Проходя в специфичной среде, свойства которой отражают жизненный опыт, эти потоки эволюционируют, угасают составляющие, противоречащие и, наоборот, усиливаются те, что согласуются с накопленными ранее знаниями и опытом. Так осуществляется естественный отбор наиболее перспективных решений, которые представляют собой комбинации стереотипов поведения, запечатленных в долговременной и оперативной памяти. Эффективность такого отбора зависит от внутренней организации нервной системы, формирование которой начинается с развитием зародыша и продолжается всю жизнь. Появляясь на свет, организм уже имеет набор жизненно важных стереотипов поведения, запрограммированных генетически или приобретенных в ходе эмбрионального развития в утробе матери. Большая часть необходимых стереотипов, формируется в первые дни и годы жизни. В частности, на этом этапе образуются стереотипы членораздельной речи, которые впоследствии могут лишь совершенствоваться, например, при освоении иностранных языков. Дальнейшее обогащение стереотипами, в частности трудовыми навыками, теоретическими знаниями происходит путем обучения, которое продолжается всю жизнь.

Говоря об интеллекте, как о способностях человеческого мозга, различают природные способности, связанные с ориентацией в новой обстановке, и способности, приобретенные при обучении или тренировке. Первые отражают генетически запрограммированный опыт, а вторые скорее относятся к внегенетической памяти, хранящейся на различных окружающих нас материальных носителях.

Заключение

Сравнивая интеллектуальные достижения, отраженные в мифах, легендах или текстах великих мыслителей древности, с достижениями интеллектуалов нашего времени, приходится констатировать, что за последние 3 тысячелетия интеллектуальный потенциал homo sapiens практически не изменился. Все достижения цивилизации получены, в основном, за счет развития материальной культуры, т.е. внегенетической памяти человечества. Изобретение письменности позволило человечеству сохранять и передавать получаемый полезный опыт в компактной символьной форме, что многократно увеличило интеллектуальный потенциал сообщества, существенно не изменив интеллектуальный потенциал самого человека. Появление компьютеров и ИНТЕРНЕТ еще более расширило коммуникационные возможности сообщества и одновременно еще более увеличило разрыв между интеллектуальными возможностями

человека и сообщества. О все возрастающем разрыве между объемом публикуемой информации и возможностями ее восприятия человеком, (информационном взрыве) ученые говорят уже более 100 лет, однако, этот рост продолжается что заставляет думать, что в данном случае мы имеем дело с законом природы, смысл которого еще предстоит понять. В эволюции природы уже известны прецеденты, когда развитие сообществ заметно опережало развитие своих индивидов. Первый такой кризис был преодолен благодаря появлению нервной системы. Более поздний прецедент это общественные насекомые – термиты, муравьи, пчелы. В таких сообществах коллективный опыт воплощается в организации их локального окружения (термитники, муравейники) и специализации поведения членов сообщества, Это тупиковые ветви эволюции, сохраняющиеся исключительно благодаря относительно стабильным условиям окружающей природы. В случае человеческой цивилизации, активно преобразующей окружающую среду, на это трудно рассчитывать.

Библиография

1. Russel S., Norvig P. Artificial Intelligence: A Modern Approach. - Prentice Hall. 1995. 932p.
2. Хайкин С. Нейронные сети
3. Коган А.Б. и др. Биологическая кибернетика. – М. Высшая школа. 1972. – 382с.
4. Прибрам К. Языки мозга. – М. Прогресс. 1975. - 464с.
5. Резник А.М. Развивающиеся системы. / в настоящем сборнике
6. Месарович М., Такахара Я. Общая теория систем. Математические основы.– М. Мир. 1978. –311с.
7. Цыпкин Я. З. Адаптация и обучение в автоматических системах. – М. Наука. 1968. –399с.
8. Різник О.М. Загальна модель розвитку. // Математичні машини і системи. -2005. -№ 1. –С. 84-98.
9. Экклз Д. Физиология синапсов. – М. Мир. 1966. 395с.
10. Ходжкин А. Нервный импульс. - М. Мир. 1965. 136с.
11. Смолянинов В.В. О некоторых особенностях организации коры мозжечка./ в сб. Модели структурно-функциональной организации некоторых биологических систем. – М. Наука. 1966. - С.203-263.
12. Horfield J. J.. Neural networks and physical systems with emergent collective computational abilities. // Proceedings of the National Academy of Science. –1982.-№79.- P. 2554-2558.
13. Резник А.М. Хопфилдовские ансамбли в латеральных нейроструктурах коры мозга. // Математичні машини і системи. – 2006.- №1.- С. 3-12.
14. Резник А.М., Городничий Д.О., Сычев А.С. Регулирование обратной связи в нейронных сетях с проекционным алгоритмом обучения. // Кибернетика и системный анализ. – 1996. - №6.- С.153-162.
15. Reznik A.M., Sitchov A.S., Dekhtyarenko O.K., and Nowicki D.W., “Associative Memories with "Killed" Neurons: the Methods of Recovery”, Proc. IJCNN, July 20-24, 2003. Portland, Oregon.
16. Шовен Р. От пчелы до гориллы. – М. Мир. 1965. – 295с.
17. Резник А.М.,. Чупаков А.Г. О структуре речевых сигналов афалины. // Бионика.- 1975.-вып. 9. – Киев, Наукова Думка, -С.125-131.
18. Кун Т.. Структура научных революций. - М. Прогресс. 1975.- 246с.
19. . Михайлов А.И, Черный А.И., Гиляревский Р.С.. Научные коммуникации и информатика. – М. Наука. 1976 – 435с.

Информация об авторе

Александр Михайлович Резник *– Институт математических машин и систем НАН Украины, зав. отделом нейротехнологий, Киев просп. Академика Глушкова 42, e-mail neuro@immsp.kiev.ua*

TWO FUNDAMENTAL PROBLEMS CONNECTED WITH AI [1]

Dimiter Dobrev

***Abstract**: This paper is about two fundamental problems in the field of computer science. Solving these two problems is important because it has to do with the creation of Artificial Intelligence. In fact, these two problems are not very famous because they have not many applications outside the field of Artificial Intelligence.*

In this paper we will give a solution neither of the first nor of the second problem. Our goal will be to formulate these two problems and to give some ideas for their solution.

***Keywords**: AI Definition, Artificial Intelligence.*

***ACM Classification Keywords**: I.2.0 Artificial Intelligence - Philosophical foundations*

Introduction

Since year 2000 we have a definition of AI [1,2,3] and since 2005 we have a program which satisfies this definition [4, 5]. Actually, these two facts are not very popular, first because the definition of AI is no accepted from almost no one except its author and second because the program which satisfies the definition of AI is useless from the practical point of view due to the combinatorial explosion.

From theoretical point of view we divide the programs in two types. The first are the non-terminating programs which will work infinitely long and the second are the terminating programs which will stop after a finite number of steps. On the other hand, from practical point of view, we divide the programs in ones that work in real time and ones which cannot work in real time. So, the fact that one program is a terminating one is useless for practical purposes if this program will work practically for infinitely long time.

That is why the program which is described in [4, 5] has no use for practical purposes and no one recognises it as AI because it does not satisfy the major requirement which is to work in real time. Even the program from [4, 5] is represented only as an algorithm. It is not written as a program because it is useless to write a program which will terminate after the end of the universe.

Therefore, if we want to make a program which will be recognised as AI we have to correct the algorithm from [4, 5] and make it work in real time. Here we have to deal with the problem of the combinatorial explosion. Even in this case the term "combinatorial explosion" is not very proper because we use this term for the cases when a programmer writes a program which should work in real time but, actually, is not working. Also, we usually assume that when we have a combinatorial explosion a faster computer can eventually help us solve the problem. In this case the situation is different. We have an algorithm which is not designed to work in real time. There is not any attempt to make the algorithm faster. The main priority has been to make the description short and clear without taking into consideration the efficiency because it is obvious that this algorithm has only theoretical value and that it will never work as a real program.

Example with the perfect compression program

So, our task is to make a real program from one algorithm which is not designed to work in real time. Actually, the algorithm in [4, 5] describes the perfect AI but we need a working AI, which does not need to be perfect.

[1] This publication is partially supported by the Bulgarian Ministry of Education (contract БОЕ 4-02/2004 г.)

We have a similar problem with the perfect compression algorithm and real compression programs. Let us define the perfect compression algorithm in order to see how little the connection between it and the real compression programs is.

Here perfect compression algorithm is called the algorithm which enumerates all programs and returns the first one (i.e. the shortest one) which generates the string which has to be compressed.

There are two things to note here. First, we have to mention that this algorithm is a non-terminating one due to the undecidability of the halting problem. In order to make it a terminating one we have to add a requirement for efficiency of the program which we search for. We can say: "the first one which generates the string for no more than ***N*** steps" but we do not want to include an additional parameter ***N*** in the definition. That is why we will say: "the program which generates the string and which has the minimal sum between its length and the number of steps which it makes while generating the string". With such correction we will obtain a compression algorithm which is a terminating one from the theoretical point of view. (Anyway, this algorithm is non-terminating in practice and therefore it is useless.)

On second place, this algorithm generates the perfect self-extracting compression file but if we assume that we have a decompression program then a shorter data file may exist, which will return our string if we input this data in the decompression program. This means that here we are talking only of self-extracting compressions.

So, we have the perfect compression algorithm. We do not say the perfect compression program because no one wrote this algorithm as a program because this is useless work. The description of this algorithm can be obtained directly from the definition of Kolmogov's complexity [10]. This means that we can say that Kolmogov is the author of the first compression algorithm but maybe this is not correct because this algorithm cannot work in real time. Today we have many programs which make compression (including self-extractable compression). These programs are not perfect but they can work in real time. Actually, these programs are much more complicated than the perfect compression algorithm and you cannot construct them directly from the perfect compressor because they are based on totally different principles.

The situation is similar with the perfect AI and the real AI. We have the perfect AI but we cannot extract a real AI which will be able to work in real time directly from it. This comes to show how difficult our task to make a real AI is.

Dividing the problem in two parts

In order to construct a real AI we will divide its work in two parts. The first part is to find a good model of the world and the second part is to choose the best action on the basis of the selected model.

Actually, in the perfect AI these two parts are not separated. We will remind that the perfect AI from [4, 5] works by trying all possible strategies in all possible models and chooses the best strategy with the biggest average result (the average result is calculated on the basis of all possible models). So, the perfect AI solves these two tasks jointly, without separating them. Nevertheless, the separation of this two problems is natural and we will make it.

If we have real time solution of both these problems then we will have a real AI. Unfortunately, both these problems lead us to a combinatorial explosion. These two problems are not very famous because they do not have many applications outside the field of the Artificial Intelligence.

We will start with the second problem which is more famous and better studied.

Finding of the correct action on the basis of a given model

We have an algorithm for solving of this problem. The name of this algorithm is **Min-Max** and we use it with great success in Chess playing programs. Nevertheless, this algorithm is not proper in all cases because sometimes it gives a combinatorial explosion. Actually, it gives combinatorial explosion even with chess but in this game we can go around the combinatorial explosion by limiting the depth in which we examine the tree of the game. This is

possible with the game of chess because we can make good evaluation of the position on the basis of things like the number of pieces on the board and on the "territory" which these pieces cover. Therefore, in some cases this problem is solvable in real time but not in all cases.

A famous example is the problem how to make a program which can play the Go game well enough to beat a professional player. A price of one million dollars was offered for working out this problem [11]. Unfortunately, the prize was not taken because the problem is too complex. The Go game looks like the chess but in it you cannot apply the **Min-Max** algorithm directly because you do not have a good evaluation function for the positions. The problem is that we have too many possible moves and mostly because in the Go game after many moves nothing essential happens (nothing which can be easily detected by a simple evaluation function).

As we said at the beginning, we will not give a solution to this problem. This is not because the One Million Dollar Prize has already expired but because we do not know how to solve this problem. Anyway, we will give some ideas. The main idea is to define intermediate goals and large steps. Actually, intermediate goal is used in the chess playing programs where this goal is to increase the value of the position. Unfortunately, this intermediate goal is given by the programmer but for AI this goal should be generated automatically because AI cannot depend on a programmer to say what is right to be done in each case.

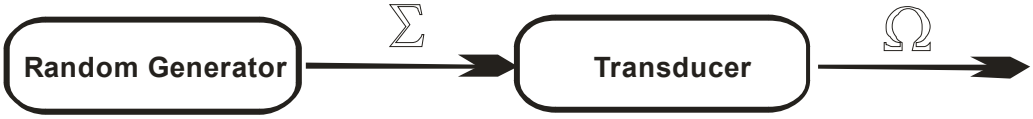
What is to think in large steps. This means to plan a chain of intermediate goals which leads to the main goal. Here we will say "goal" for events which we evaluate as good ones. One event can be evaluated as a good one by apriory or because it is part of a chain which leads to an event which has already been evaluated as a good one. So, thinking by large steps will be planing chains of events. For such planing we can use the **Min-Max** algorithm but here the problem is how to define events automatically and how to automatically find the way for transition from one event to another. For example, with the game of chess you have events "taking of enemy piece" and "winning the game". There is a connection between these two events and this connection is built in the chess playing programs by their creators. So, the chess playing program tries to take enemy pieces in order to win the game. The problem is how to make a program which defines events automatically and automatically evaluates these events as good or bad. Also, AI has to be able to automatically find connections between these events in order to plan a chain of events.

Actually, all these thoughts lead us to the fact that in order to solve the second problem we need a solution of the first one because in order to think in large steps we need an automatic detection of events and this is part of the problem of finding a good model of the world. From this point on we will talk only about the solution of the first problem.

Formalisation of the first problem

Here when we say a good model we mean an adequate one. So, this means a model which will give a correct predictions for the future.

The first step in solving a problem is to formalise it. Let us examine the following formal problem. Let us have a two finite alphabets Σ and Ω. Let us have a random generator which generates letters from the alphabet Σ and a transducer which inputs a letter from Σ and outputs a letter from Ω. The goal is to built a model of this transducer which will give us the possibility to guess its next output if we know the input letter and if we know the entire history (i.e. if we know the row $a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}, a_n$ where a_i are the letters from the random device which are inputted in the transducer and b_i are the letters which are outputted.) So, the question is what will be the output on the step n if we know all data from step **0** to step $n-1$ and the input on the step n. In other words, what will be b_n if we know $a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}, a_n$.



What is the connection between this formalisation and the definition of AI [1,2,3]? Here we have a random generator and transducer, which interact. In the definition of AI the transducer corresponds to the concept of World. Here we try to make a model of the transducer but in [1,2,3] AI tries to understand the World. This means that here the random generator corresponds to the AI from [1,2,3]. Where is the difference? AI reads the output of the World (of the transducer) but the random generator does not have any input. AI is able to carry out some experiments in order to understand the World but the random generator does not make any intentional experiments. Anyway, if the observer waits long enough the random generator will make all experiments (accidentally).

Note: In [1,2,3] the alphabets Σ and Ω are Ω and Σ and the letters ***a*** and ***b*** are ***d*** and ***v***. This can cause confusion in understanding the connection between [1,2,3] and this paper.

How to find a good model of the world (transducer)

So, we have a formalisation and now our problem is formal. As we said this problem is not famous because it has not many applications outside the field of AI. It is even difficult to find an example for a practical problem which leads to this theoretic formalisation. The only such example which we have in mind is the following. Let us have a program protected against illegal use by a hardware key device. If we want to break this protection we have to understand how this hardware key works and try to recreate it. Really, the practical problem allows us to open the key and see how it is designed but here we assume that we have no such possibility and that we have to observe the key as a black box and to study only its input and output.

As you see, there are not many applications of this problem. Maybe this is the reason that nobody offers a price for its solution but nevertheless, here we will discus this problem.

So, is this problem solvable. In the general case the answer is no because if we do not make any suggestions about the transducer then we will not be able to say anything about its next output. For example, if it outputs one and the same letter one hundred times in a row irregardless of the input then we can predict that on the next step it will workout the same letter. This prediction looks natural but it lies on the conjecture that the simpler explanation is more probable than the complicated one. Without this conjecture we cannot make any prediction because it is possible that in this case we have a transducer which outputs one hundred times one and the same letter and on the next step it outputs another letter.

We said that we will look for the simplest model of the transducer. Also, we have to bear in mind that we need a solution which works in real time.

Another question is whether our transducer is deterministic device or not. It will be much easier if we assume that the transducer is a deterministic device but if we restrict our search only in the set of deterministic models then the chance to find a proper model in a concrete situation is very small.

Next question. How many internal states our transducer has. It is reasonable to suggest that the number of internal states is finite (i.e. that it is finite automata). Anyway, the more general case is to suggest countably many internal states. It is no use suggesting an uncountable number for the internal states because only a countable subset of them will be obtainable in the deterministic case. In the non-deterministic case there is some use in suggesting an uncountable number of internal states but if we restrict our observation to the set of calculable functions then again there is no use suggesting an uncountable number for the internal states of the transducer.

The last question. Is our transducer a calculable function or not. Definitely yes. We are looking for a practical solution so it has to be a calculable function and even it has to be an easy calculable function (i.e. calculable for small number of steps without problems like combinatorial explosion). Beside that, every non-calculable function can be approximated with a calculable one (of course, until the concrete moment ***n*** but not until the infinity).

One theoretical solution

Here we will give the next useless theoretical solution which cannot work in real time. The reason that we give this solution is to show that such one exists. This is important because we cannot give a solution which can work in real time. Instead of that at the end of this paper we will give some ideas about the creation of real time solution.

Here is our theoretical solution. First for the deterministic case:

It will enumerate all programs and will return the first one (i.e. the shortest one) which generates the row $b_0, ..., b_{n-1}$ if the input is $a_0, ... , a_{n-1}$. Here we have a problem with undecidability of the halting problem again. So, we will take not the shortest one but this which has minimal sum between its length and the maximum number of program steps which it needs to generate any of the outputs (i.e. any of $b_0, ..., b_{n-1}$). So, this algorithm will give us a short and quick program which makes a very good prediction of b_n. The only problem is that we will have to wait this algorithm to finish almost forever.

For the non-deterministic case we have to complicate our algorithm a little bit.

First we will complicate our programs (which we use as models) by adding one subroutine **random( )** which will return zero or one with possibility **1/2**. With this subroutine we cannot generate even the possibility **1/3** but by using subroutine **random( )** we can approximate any possibility (nevertheless is it rational or irrational number).

Now, when we deal with non-deterministic models we cannot say simply yes or no to the question does this model generate our sequence or not. Instead of that, we can calculate the possibility for our sequence to be generated. Of course, here we will have the problem with the non-terminating models again and in order to keep things calculable we will add one constant ***Max*** and we will calculate the possibility of the model to generate our sequence for no more than ***Max*** steps per output.

What is the prediction of one non-deterministic model for b_n. First we do not know what is the internal state of the model when it inputs a_n because there may be more than one possible way for this model to generate $b_0, ..., b_{n-1}$. Even if we know the internal state we cannot say which letter will be worked out as b_n because our model is non-deterministic. Nevertheless, we can calculate for concrete model the possibility for every letter to be worked out.

Every model will give us some prediction but we have to choose which one to trust and which prediction to accept as the better one. This question will not be discussed in this paper.

Some ideas about the practical solution

First, in order to make real time solution we will restrict the observation to the set of models with a finite number of states (finite automata). Of course, this restriction is essential because some of the worlds (transducers) cannot be described with a finite models. Anyway, in many cases the finite models are sufficient or at least they can give a good approximation of the World. (You can find in [7] the idea that we can raise the finite models with first order axioms in order to make models for more complex worlds.)

Second, we have to mention that we will look for a set of good models instead of a single model. The chance to find a single model which describes the world is small. It is more probably to find many different simple models which describe different features of the world. Also, in this way our system will be more consistent because in its life (work time) it will change some of the selected models instead of changing the only model which can make its behaviour totally different.

Now, let us start with the case of deterministic models. Such model looks like a deterministic finite automata (with finite number of states, starting state, arcs labelled with the letters from Σ, etc.) but here we will have only one type of states (no final states) and we will have a second label on every arc which will be a letter from Ω.

If we have such a model with a reasonable number of states we can easily find it by a backtracing algorithm similar to the one from [8]. Anyway, the existence of such model is very suspicious because if we have

deterministic model then we will be able 100% correctly to predict the future. This will mean that the world is very simple, which is not the interesting case.

Let us look for a non-deterministic model of the world. Actually, as we said, we will look for a set of many non-deterministic models.

We will divide the non-deterministic models in two groups - partially deterministic and totally non-deterministic. Examples of these two types of non-deterministic models are found in [6, 7].

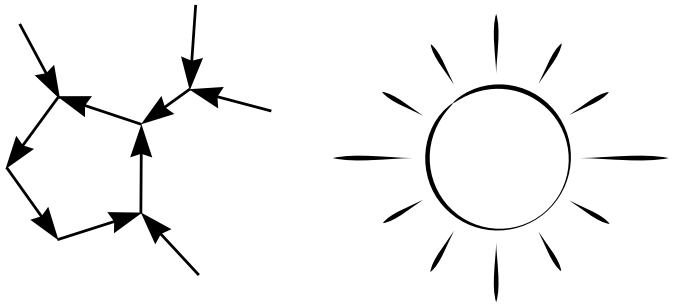
Partially deterministic models will look like the deterministic models but with the difference that they will have a second label on the arcs, which is a set of letters from Ω instead of one letter. Actually, this will be a set of 2-tuples from letter and possibility because every letter from Ω will have its own possibility to be worked out in the case when the model is in the respective state and the input letter from Σ is that which is the first label of the arc.

The good side of partially deterministic models is that their current state is determined. From every deterministic automata on alphabet Σ we can make partially deterministic model by defining the possibilities through statistics on the basis of life experience ($a_0, b_0, \ldots, a_{n-1}, b_{n-1}$). In fact, statistics will not give us the possibility but the number of times certain letter is worked out in certain situation. In this case (1, 1) is different from (30, 30) because in both cases we have 50% possibility but in the second case this is more certain.

So, we have so many partially deterministic models and the question is which of them are better. This is a very difficult question and we will not discus it here. We will say only that if one model gives in some cases (i.e. arcs) prediction which is useful (for example 100% possibility for certain letter) and reliable (i.e. this link is used many times) then this model is useful.

How to find a good partially deterministic model. First we need a definition which strictly says which model is better. The second problem is that we have to search for this model in huge set of possible candidates.

The idea which we will give in this paper is to observe the behaviour of a single letter. We will call this method the "sunshine" method for finding finite automata. This idea is based on the fact that if we observe only the arcs which have a certain letter as a first label these arcs make one or more figures which we will call "Suns". The Sun is a cycle with paths which flow in it. This figure looks like the picture of the sun which children use to draw.



The idea is that we will be able to relatively easy detect the dependency in the figure Sun and after constructing several suns to construct the finite automate from these suns. In order to catch dependencies for one letter we will need to observe long sequences of this letter. We may wait long until the random generator generates such sequence (especially if the alphabet Σ is big which is the general case). That is why we will use elimination of letters and construction of compound letters. Elimination of letters is when we assume that some letters do not change the state of the model. Compound letters are sets of letters which we accept as one letter. In [6, 7] we have an example of partially deterministic model where we use letters "left" and "right". There we assume that all other letters do not change the state of the model (i.e. these letters are eliminated). In the next model in [6, 7] we have the letter "victory or loss" which is compound. Actually, this compound letter is not from Σ but from Ω. Really, in the deterministic model there is no sense to include the output of the transducer as information our model depends on but in a non-deterministic case this information is essential and it is reasonable to use it in our model.

Bibliography

[1] Dobrev D. D. AI - What is this. In: PC Magazine - Bulgaria, November'2000, pp.12-13 (in Bulgarian, also in [9] in English).
[2] Dobrev D. D. AI - How does it cope in an arbitrary world. In: PC Magazine - Bulgaria, February'2001, pp.12-13 (in Bulgarian, also in [9] in English).
[3] Dobrev D. D. A Definition of Artificial Intelligence. In: Mathematica Balkanica, New Series, Vol. 19, 2005, Fasc. 1-2, pp.67-74.
[4] Dobrev D. D. Formal Definition of Artificial Intelligence. In: International Journal "Information Theories & Applications", vol.12, Number 3, 2005, pp.277-285.
[5] Dobrev D. D. Formal Definition of AI and an Algorithm which Satisfies this Definition. In: Proceedings of XII-th International Conference KDS 2006, June, 2006 Varna, Bulgaria, pp.230-237.
[6] Dobrev D. D. Testing AI in One Artificial World. In: Proceedings of XI-th International Conference KDS 2005, June, 2005 Varna, Bulgaria, pp.461-464.
[7] Dobrev D. D. AI in Arbitrary World. In: Proceedings of 5th Panhellenic Logic Symposium, July 2005, University of Athens, Athens, Greece, pp. 62-67.
[8] Dobrev D. D. First and oldest application, http://www.dobrev.com/AI/first.html
[9] Dobrev D. D. AI Project, http://www.dobrev.com/AI/
[10] Kolmogorov A. N. and Uspensky V. A. Algorithms and randomness. - SIAM J. Theory of Probability and Its Applications, vol. 32 (1987), pp.389-412.
[11] The Million Dollar Prize http://www.reiss.demon.co.uk/webgo/million.htm

Author's Information

Dimiter Dobrev – *Institute of Mathematics and Informatics, BAS, Acad.G.Bonthev St., bl.8, Sofia-1113, Bulgaria; P.O.Box: 1274, Sofia-1000, Bulgaria; e-mail: d@dobrev.com*

РАЗМЫШЛЯЮЩИЕ КОМПЬЮТЕРЫ

Виталий Ященко

Аннотация: *В статье рассматривается концепция создания интеллектуальных компьютеров и систем, познающих и размышляющих на основе моделирования нейрофизиологических свойств мозга с использованием новой технологии обработки информации. Новая технология обработки информации (нейроподобные растущие сети) разработана на основе анализа и синтеза знаний, выработанных физиологами и различными направлениями Computer science, заключенных в технологиях обработки информации в биологических объектах, семантических сетях, нейронных сетях и интеллектуальных системах. Нейроподобные растущие сети это новый тип нейронных сетей, которые представляют собой растущую, динамическую структуру, изменяющуюся в зависимости от значения и времени поступления информации на рецепторы, а также предыдущего состояния сети. Предлагаемая концепция, которая объединяет физический и виртуальный миры, имеет универсальный характер. Такой подход дает новое основание для развития и массового производства продвинутых размышляющих компьютеров, интеллектуальных систем и роботов. Последние могут иметь разнообразные важные применения в гражданской и военной областях особенно для выполнения действий в непредсказуемых ситуациях и опасных окружающих средах.*

Ключевые слова: *мозг, сознание, искусственное сознание и подсознание, интеллектуальные системы, роботы, семантические растущие сети, нейронные сети, нейроподобные растущие сети, частичное возбуждение нейроподобных элементов.*

Введение

Проблема искусственного интеллекта (ИИ) возникшая в 30-40 годах прошлого столетия, является одной из актуальнейших проблем современности. Вопросы, является ли интеллект исключительно привилегией человеческого мозга, возможно ли создать техническое устройство способное размышлять, обладать разумом до сих пор волнуют умы ученых.

Моделирование процесса мышления на нейронных сетях (НС)

К наиболее изученному классу НС относят персептроны. Персептрон применяется для задач автоматической классификации, которые в общем случае состоят в разделении пространства признаков между заданным количеством классов [Rosenblatt R.,1962].

Карты Кохонена обладают свойством сохранения топологии, которое воспроизводит важный аспект карт признаков в коре головного мозга высокоорганизованных животных. Сеть успешно применяется для распознавания речи, обработки изображений, в робототехнике и задачах управления [Kohonen T.,1989].

В сети Хопфилда используется одноуровневая структура ассоциативной памяти, в которой выходной вектор появляется на выходе тех же нейронов, на которые поступает входной вектор [Богомолова В., 2003].

В нейронной сети ART (Artificial Resonance Theory) моделируются механизмы кратковременной и долговременной памяти [Гроссберг С.,1997]. Различными типами сетей ART решаются задачи распознавания зрительных образов, обработки потоков звуковой информации, распознавания речи, моделирования управлением движения глаз и представления информации в соматосенсорной коре.

По мнению автора для решения проблемы создания размышляющих компьютеров моделирование отдельных психофизиологических свойств мозга, как отдельных компонент системы, с последующими попытками их объединения, малоэффективны. Необходимо, опираясь на исследования нейрофизиологов, разработать ассоциативную вычислительную структуру как аппарат реализации психофизиологических свойств мозга человека.

Для реализации функций мышления в интеллектуальных компьютерах, автором предлагается новый тип НС - многомерные рецепторно – эффекторные нейроподобные растущие сети (мрэн-РС). В основу мрэн-РС положены растущие пирамидальные сети разработанные профессором В.П. Гладуном.

Многомерные нейроподобные растущие сети

Многомерные рецепторно-эффекторные нейроподобные растущие сети формально задаются следующим образом:

$$S = (R, A_r, D_r, P_r, M_r, N_r, A_e, D_e, P_e, M_e, E, N_e),$$

$R \supset R_v, R_s, R_t$, $A \supset A_v, A_s, A_t$, $D \supset D_v, D_s, D_t$, $P \supset P_v, P_s, P_t$, $M \supset M_v, M_s, M_t$, $N \supset N_v, N_s, N_t$, $E \supset E_r, E_{d1}, E_{dn}$, $A \supset A_r, A_{d1}, A_{dn}$, $D \supset D_r, D_{d1}, D_{dn}$, $P \supset P_r, P_{d1}, P_{dn}$, $M \supset M_r, M_{d1}, M_{dn}$, $N \supset N_r, N_{d1}, N_{dn}$,

здесь R_v, R_s, R_t - конечное подмножество рецепторов; A_v, A_s, A_t - конечное подмножество нейроподобных элементов рецепторной зоны; D_v, D_s, D_t - конечное подмножество дуг (связей) рецепторной зоны; P_v, P_s, P_t - конечное множество порогов возбуждения нейроподобных элементов рецепторной зоны, принадлежащих, например, визуальному, звуковому, тактильному информационным пространствам; N_r- конечное множество переменных коэффициентов связности рецепторной зоны; E_r, E_{d1}, E_{dn} - конечное подмножество эффекторов; A_r, A_{d1}, A_{dn} - конечное подмножество нейроподобных элементов эффекторной зоны; D_r, D_{d1}, D_{dn} - конечное подмножество дуг (связей) эффекторной зоны; P_r, P_{d1}, P_{dn}- конечное множество порогов возбуждения нейроподобных элементов эффекторной зоны, принадлежащих, например, речевому информационному пространству и пространству действий; N_e - конечное множество переменных коэффициентов связности эффекторной зоны.

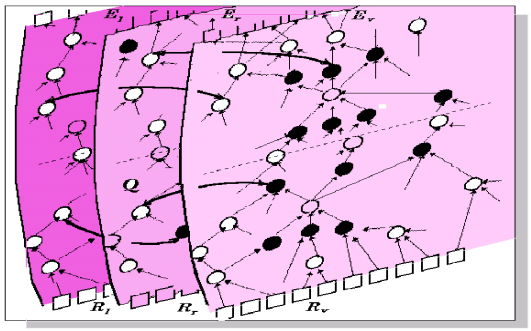


Рис.1. Многомерная рецепторно-эффекторная нейроподобная РС

Нейроподобные растущие сети являются динамической структурой, которая изменяется в зависимости от значения и времени поступления информации на рецепторы, а также предыдущего состояния сети. В н-РС информация о понятиях, объектах и ситуациях представляется ансамблями возбужденных нейроподобных элементов и связями между ними. Формируется совокупность устойчивых связей описываемого объекта, обеспечивающих его целостность и тождественность самому себе. Восприятие описаний объектов и ситуаций сопровождается вводом в сеть новых нейроподобных элементов и дуг при переходе какой-либо группы рецепторов и нейроподобных элементов в состояние возбуждения, т.е. в процессе восприятия информации сеть перестраивает свою структуру, запоминая, классифицируя и обобщая эту информацию. Рецепторно-эффекторные н-РС, содержащие рецепторные и эффекторные зоны, позволяют на соответствующие условия (восприятие информации) вырабатывать управляющие воздействия во внешний мир и формировать поведение системы. В рецепторной зоне осуществляются накопление и реорганизация уже существующих моделей (нейронных ансамблей) адекватных условиям, возникающим во внешней среде, а в эффекторной зоне вырабатываются, накапливаются и реорганизуются модели действий, адекватных внешним условиям, и, таким образом, осуществляется активное взаимодействие с окружающей средой.

В многомерных рецепторно-эффекторных нейроподобных растущих сетях (рис.1) вышеописанные процессы происходят одновременно во всех информационных пространствах, запоминая, классифицируя и обобщая воспринимаемую и вырабатываемую информацию в визуальном, звуковом, тактильном и др. представлениях [Ященко В.,1995], [Yashchenko V.,1999] .

Моделирование функции мышления на мрэн-РС

Модели функций «мысль» и «мышление» на многомерных рецепторно – эффекторных нейроподобных сетях описываются следующим образом:

модель функции «мысль» – ансамбль возбужденных нейроподобных элементов (внутренняя модель внешнего или абстрактного мира), усиленная функцией мотивации в данный момент;

модель функции «мышление» – размышление, последовательность мыслей – последовательное взаимодействие ансамблей возбужденных нейроподобных элементов (внутренних моделей), направляемое уровнями возбуждения нейроподобных элементов, усиленными или ослабленными функцией мотивации.

Рассмотрим пример.

Предположим, что компьютер, система или интеллектуальный робот обладают структурой мрэн-РС, в которой содержатся знания в виде символов, слов и предложений следующего содержания:

а) Сократ человек;

б) Платон человек;

в) Человек смертен;

г) Бог бессмертен;

д) Зевс Бог.

Как будет «мыслить» компьютер, размышляя над вопросом, «*Сократ смертен?*» (рис.2).

При поступлении вопроса на рецепторное поле в рецепторной зоне возбуждаются нейроподобные элементы, соответствующие понятиям «Сократ» и «Смертен». Частично возбуждаются нейроподобные элементы, соответствующие понятиям «Сократ человек», «Платон человек» и «Человек смертен». В эффекторной зоне частично возбуждены нейроподобные элементы, соответствующие понятиям «Сократ человек», «Платон человек» и «Человек смертен» и, усиленные функциями внимания и мотивации, полностью возбуждены командные нейроподобные элементы, соответствующие понятиям «Сократ», «Человек» и «Смертен». Ответ компьютера - *Сократ человек смертен.*

Посмотрим теперь, как будет «мыслить» компьютер, размышляя над вопросом, «*Зевс смертен?*» (рис.3-5). При поступлении вопроса на рецепторное поле, в рецепторной зоне возбуждаются нейроподобные элементы, соответствующие понятиям «Зевс» и «Смертен». Частично возбуждаются нейроподобные элементы, соответствующие понятиям «Зевс бог», «Сократ человек», «Платон человек» и «Человек смертен». В эффекторной зоне, усиленные возбуждениями функций внимания и мотивации, частично возбуждены нейроподобные элементы, соответствующие понятиям «Зевс бог» и «Человек смертен» и возбуждены командные нейроподобные элементы, соответствующие понятиям «Зевс», «Смертен» и частично возбуждены «Бог» и «Человек».

Компьютер на уровне «внутреннего проговаривания» (мысленного проговаривания) произнесет - Зевс Бог человек смертен. Таким образом, на рецепторное поле поступает фраза «Зевс Бог человек смертен» (рис.4). Теперь в рецепторной

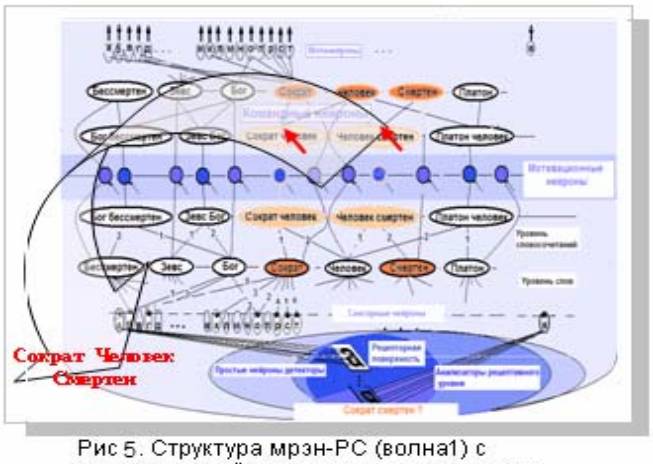


Рис 5. Структура мрэн-РС (волна1) с активными нейроподобными элементами

Рис.2. Структура мрэн-РС (волна 1)

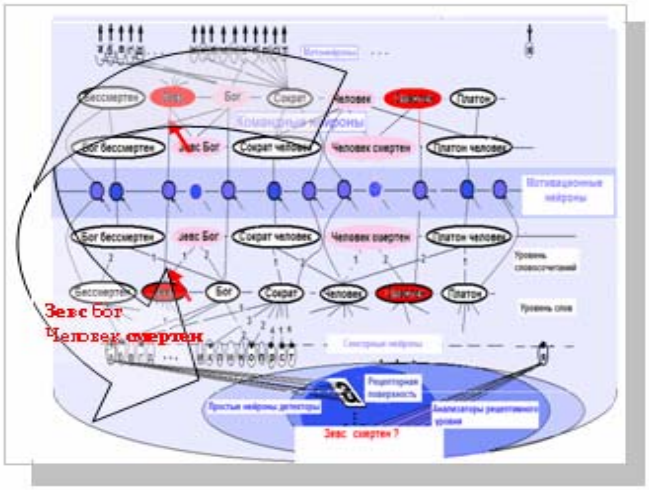


Рис.6. Структура мрэн-РС (волна 1) с активными нейроподобными элементами

Рис.3. Структура мрэн-РС (волна 1)

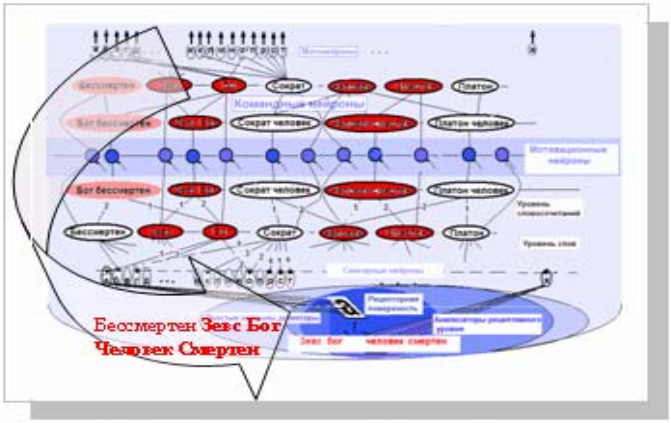


Рис.7. Структура мрэн-РС (волна 2) с активными нейроподобными элементами

Рис.4. Структура мрэн-РС (волна 2)

зоне возбуждаются нейроподобные элементы, соответствующие понятиям «Зевс», «Бог», «Человек», «Смертен», «Зевс Бог», «Человек смертен».

Частично возбуждаются нейроподобные элементы, соответствующие понятию «Бог бессмертен». В эффекторной зоне частично возбуждены нейроподобные элементы, соответствующие понятию «Бессмертен», и возбуждены командные нейроподобные элементы, соответствующие понятиям «Зевс», «Бог», «Человек», «Смертен». Компьютер на уровне «внутреннего проговаривания» произнесет - Бессмертен Зевс Бог человек смертен. Таким образом, на рецепторное поле поступает фраза «Бессмертен Зевс Бог человек смертен» (рис.5). Теперь в рецепторной зоне возбуждаются нейроподобные элементы, соответствующие понятиям «Бессмертен» «Зевс», «Бог», «Человек», «Смертен», «Зевс Бог», «Человек смертен», «Бог бессмертен».

Так как командные нейроны, соответствующие понятиям «Зевс», «Бог», «Человек», «Смертен», возбуждены полностью, компьютер произнесет - Зевс Бог, Бог бессмертен, Человек смертен. Ответ компьютера - Зевс Бог, Бог бессмертен, Человек смертен.

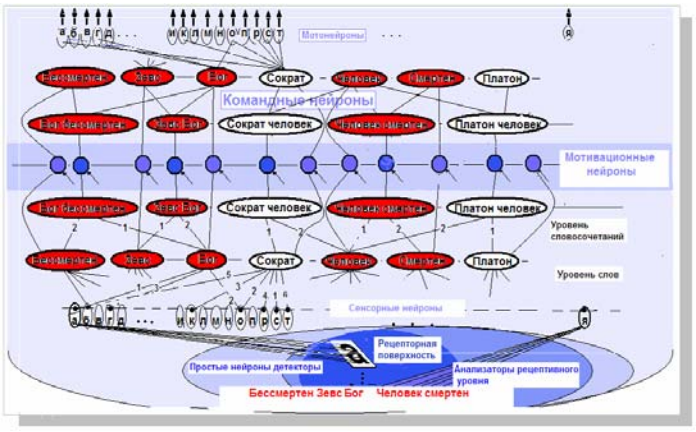


Рис.5. Структура мрэн-РС (волна 3)

Наличие обратной связи в виде «внутреннего проговаривания» позволяет системе на внешнее воздействие (постановка вопроса) осуществить несколько циклов передачи внутренней активной информации на вход системы, осуществлять циклы «восприятие – действие» за счет распространения волн возбуждения по ансамблям нейроподобных элементов, командных нейронов и мотонейронов до достижения цели.

Предложенная концепция проверена на программной модели “Think”. Модель разработана в среде Borland C++ Builder 5.0 под платформы Windows 95/98/NT/2000. Интерфес программной модели “Think” представлен на рис.6.

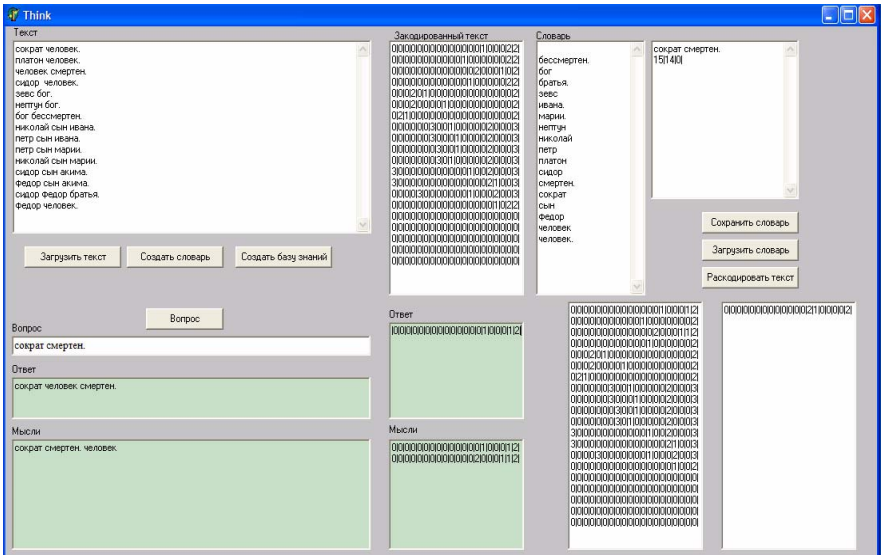


Рис.6. Интерфейс програмной модели "Think"

Заключение

Конечно, это весьма упрощенная модель функции мышления. Здесь не показано, каким образом кодируется визуальная и звуковая информация в соответствующих информационных пространствах и как она участвует в процессе мышления. Здесь также не рассмотрены вопросы образного мышления, его связь и взаимодействие с мышлением логическим, т.е. не рассмотрены модели механизмов работы и взаимодействия левого и правого полушарий мозга. Но даже при таком упрощенном варианте модели мышления мы получаем правильный ответ.

Считается, что основным априорным условием сознательного состояния является способность обозревать свое внутреннее состояние посредством периодической передачи на вход имеющейся в памяти информации. Повторный ввод хранящейся в памяти информации позволяет распознавать ее и сравнивать с содержимым памяти. Тем самым осуществляется просмотр формируемых внутри образов (моделей) в потоке внешней информации. Следовательно, процесс осознания представляет собой ассоциативное воспоминание с обновлением и требует периодического распознавания информации, представляющей внутреннее состояние (образ) и внешнюю среду (реальный мир) [Эделмен Дж., Маунткаса В.,1981], [Середин В.,1987].

В итоге можно сделать следующее заключение. Мыслить, размышлять, значит, сознавать. В этом смысле «внутреннее проговаривание» - циклы передачи внутренней активной информации на вход системы – можно рассматривать как модель искусственного сознания интеллектуального компьютера, а циклы передачи внутренней активной информации на вход системы без включения проговаривания рассматривать как модель искусственного подсознания.

Таким образом, система, обладающая механизмами *искусственного сознания и подсознания,* получает возможность *рассуждать, упорядочивать и корректировать свои знания.* Осуществляя повторный неоднократный ввод хранящейся в памяти информации, снова распознавая ее и сравнивая с содержимым памяти, тем самым, выполняет неоднократный просмотр и коррекцию формируемых внутри образов (моделей внешнего мира) в непрерывном потоке информации реального внешнего мира. Действительно, по существу, процесс осознания представляет собой ассоциативное воспоминание с обновлением и требует периодического распознавания информации, представляющей внутреннее состояние (образ) и внешнюю среду (реальный мир).

Литература

[Богомолова В. ,2003] Богомолова В. В мире кричащего БЕЗМОЛВИЯ.htm.

[Rosenblatt R.,1962] Rosenblatt R. Principles of Neurodynamics. Spartan Books.- New York.- 1962. – p.174

[Kohonen T.,1989] Kohonen T. SelfOrganization and Associative Memory // Third Edition.- Springer-Verlag.- N.York, 1989.

[Гроссберг С.,1997] Гроссберг С. Внимательный мозг// Открытые системы. - 1997.- №4.- С. 29-33.

[Ященко В.,1995] Ященко В. Рецепторно-эффекторные нейроподобные растущие сети - эффективное средство моделирования интеллекта.II // Кибернетика и сист. анализ 1995. - №5.- С. 48-57.

[Yashchenko V.,1999] Yashchenko V. Receptor-effector neural-like growing network - an efficient tool for building intelligence systems / Proceedings of the second internatiional conference on information fusion, Sunnyvale Hilton Inn, Sunnyvale, California, USA: 1999.- July 6-8, Vol.II. - P. 1113-1118.

[Эделмен Дж., Маунткаса В.,1981] Эделмен Дж., Маунткаса В. Разумный мозг. - М.: -1981. -133 с.

[Середин В.,1987] Середин В. Мозг как вычислительная система // Информатика и образование. - 1987 г. - №6.

Информация об авторе

Ященко В.А. *Институт проблем математических машин и систем Кибернетического центра имени Глушкова В.М. НАН Украины, Киев, старший научный сотрудник.* *Email:* mis@immsp.kiev.ua

МОДЕЛИРОВАНИЕ СУБЪЕКТИВНОГО ПРЕДСТАВЛЕНИЯ ЧЕЛОВЕКА О ДЕЙСТВИТЕЛЬНОСТИ

Алексей Бычков, Михайл Меркурьев

***Аннотация**: Предлагается математический формализм для описания субъективного представления о действительности (perception) и нечеткого поведения и нечетких процессов. За основу берется теория возможности предложенная Заде и развитая, в дальнейшем Пытьевым Ю.П. Для описания нечетких процессов предлагается использовать новый класс дифференциальных уравнений.*

***Ключевые слова**: персептивный элемент, персептивное множество, продолжение меры на булеан, процесс нечеткого блуждания, нечеткая устойчивость.*

***ACM Classification Keywords**: G.1.7, G.3*

Вступление

Любое моделирование есть работа естественного интеллекта по познанию окружающей среды. Существует множество определений естественного интеллекта [1-4 и др.]. Платон определяет интеллект как то, что отличает человеческую душу от животных. В средневековой западноевропейской схоластике это понятие употреблялось для обозначения высшей познавательной способности в противоположность разуму. В немецкой классической философии термином «интеллект» обозначали способность образования понятий. В дальнейшем интеллект рассматривается как врожденная или благоприобретенная способность человека к познанию, мыслительная способность человека.

Обобщая их, можно сказать, что *естественный интеллект* (ЕИ) есть способность к приобретению, хранению, преобразованию, обобщению данных и к самоадаптации к изменениям окружающей среды.

Но следует к этому обобщению добавить еще «при недостатке и неточности поступающей и имеющейся информации». Это позволит ЕИ принимать решения, касающиеся того, что еще не произошло, т.е. рассматривать различные *возможности* будущего и сравнивать их.

Таким образом, ЕИ — это способность позволяющая моделировать события и объекты реального и виртуального миров (сущности). Для такого моделирования необходимо:

1. *язык описания данных* (определения, понятия);
2. язык манипулирования данными *(аксиомы, правила вывода);*
3. *проверка на непротиворечивость*.

Язык является необходимым условием существования интеллекта, ибо основной единицей информации в ЕИ является *понятие* «представление, отчетливое осознание которого связано со словом» [4]. Для ЕИ именно использование слов и языка дает возможность связывать понятия в логические выводы, т.е. получать новые знания, понятия — модель сущности.

Манипуляция с данными включает в себя:

1. *абстракцию и спецификацию*;
2. *получение новых (составных понятий) из уже имеющихся*;

Манипулирование данными происходит после субъективного восприятия естественным интеллектом сущности. Субъективизм ЕИ проявляется в его эмоциональной составляющей.

Наличие эмоциональной составляющей у ЕИ, в свою очередь, позволяет внести в процесс моделирования то, что называют «человеческим фактором», т.е. ЕИ способен получить ε-адекватную модель неточного объекта или процесса. Например, для описания результата процесса измерения

температуры воды ЕИ может использовать слова «очень + холодная», «холодная», «теплая», «горячая», «сильно + горячая».

Однако не надо забывать об интуиции. Ее можно рассматривать как работу интеллекта при неполных, расплывчатых данных, т.е., естественный интеллект в виртуальной реальности.

Обобщая, можно сказать, что без эмоций, языка и интуиции ЕИ — только множество условных и безусловных рефлексов и их приобретение. Или рассматривая понятия эмоций, языка, интуиции, приобретение рефлексов как некие рефлексы, естественный интеллект — это совокупность рефлексов, которые вырабатываются в результате взаимодействия носителя интеллекта с окружающей средой или внутри себя, возможно при неточных, расплывчатых данных. Это взаимодействие, на самом деле, является познанием окружающего мира и самого себя.

При познании (моделировании) ЕИ получает информацию от познаваемой сущности и от других ЕИ. При этом ЕИ пользуется языком передачи и обработки информации.

Как уже отмечалось, в ЕИ присутствует эмоциональная составляющая. Начал формализовать это присутствие Л. Заде, введя понятие нечеткого множества [5,6]. Он вводит это понятие для формализации субъективных суждений [7]. Заде определяет нечеткое множество посредством задания универсального множества X для описания предметной области и множества пар $\{x_i \in X, \mu(x_i)\}$ для описания степени адекватности элемента x_i формализуемому суждению. Другими словами, $\mu(x_i)$ — степень доверия к элементу x_i как к описанию суждения. В этом и проявляется субъективизм. В теории Заде нечетким числом называется нечеткое множество, заданное на R^1, т.е., $X = R^1$.

При такой терминологии нельзя рассматривать (в отличие от классической теории множеств) нечеткое множество как совокупность нечетких элементов. Нечеткие множества Заде позволяют лишь субъективное представление ЕИ о познаваемой сущности.

Для формализации отношения ЕИ к познаваемой сущности необходимо ввести функцию меры отношения [7,9]. Пусть X — как и прежде, универсальное множество, необходимое для формализации предметной области. Введем функцию $m(\cdot)$, которая описывает степень уверенности имеющейся у ЕИ по отношению к сущности A, $A \subset X$.

Эта функция меры должна отвечать естественному свойству монотонности $\forall A, B \subset X$: $A \subseteq B \Rightarrow m(A) \le m(B)$. Если знак $\subseteq$ интерпретировать как «сущность A является причиной сущности B», то свойство монотонности можно интерпретировать как «уверенности ЕИ в сущности B, по крайней мере, столько же, сколько уверенности в сущности A».

Из монотонности непосредственно следует, что $\forall A, B \subset X$ имеем

$$m(A \cup B) \ge \max(m(A), m(B)),\ m(A \cap B) \le \min(m(A), m(B)).$$

Частным случаем этих неравенств будут функции, для которых

$$P(A \cup B) = \max(m(A), m(B)),\ N(A \cap B) = \min(N(A), N(B)).$$

Функции P и N называют мерами возможности и необходимости [7,11]. Они связаны соотношением

$$P(A) = 1 - N(\overline{A}).$$

В эксперименте с измерением температуры воды функция P — это отношение ЕИ к результатам измерения, т.е., из двух результатов выбирается тот, уверенность в котором больше. ЕИ абсолютно уверен в том значении $N = 1$, для которого противоположное невозможно ($P = 0$).

Функции возможности и необходимости удовлетворяют таким соотношениям: $\max(P(A), P(\overline{A})) = 1$, $\min(N(A), N(\overline{A})) = 0$, что существенно отличает такое задание отношения от

вероятностного подхода, при котором вероятность некоторого события полностью определяется вероятностью противоположного события

$$P(A) = 1 - P(\overline{A}),$$

тогда как возможность и необходимость противоположных событий связаны соотношениями

$$N(A) + N(\overline{A}) \le 1, P(A) + P(\overline{A}) \ge 1.$$

Однако введение таких функций еще не позволяет полностью формализовать эксперимент с измерением температуры воды.

Напомним, что эксперимент имеет три составляющие: воду, прибор для измерения и значение измерения. Мы формализовали только реальную температуру воды с помощью универсального множества, субъективное восприятие значения измерения с помощью нечеткого множества и отношение к этому измерению.

Для полной формализации необходимо еще задать функцию, описывающую работу прибора и функцию для описания неточности полученных прибором значений температуры воды.

Поэтому рассмотрим применение теории возможностей для описания нечетких величин [11-13].

Элементы теории возможностей

Пусть $(X, \mathbf{A})$ – измеримое пространство, $\mathbf{A}$ — σ-алгебра на X. Через $\beta(X)$ будем обозначать совокупность всех подмножеств множества X. Предлагаемый подход к описанию субъективности базируется на модели предложенной в [11].

Определение 1. [11] Шкалой значений возможностей называется полукольцо $L = \{[0,1], \le, +, \bullet\}$, то есть, отрезок [0,1] с обычным порядком $\le$ и двумя операциями:

1. $a + b = \max\{a, b\}$;
2. $a \bullet b = \min\{a, b\}$.

В дальнейшем будем рассматривать $\mathbf{A}$-измеримые функции $f : X \to L$. Обозначим через $L(X)$ класс таких функций, для которого выполняются следующие свойства [11]:

1. $f \in L(X), a \in L \Rightarrow a \bullet f = \min(a, f(x)) \in L(X)$;
2. $f, g \in L(X) \Rightarrow f + g = \max(f(x), g(x)) \in L(X)$; $f \bullet g = \max(f(x), g(x)) \in L(X)$;
3. $f \in L(X) \Rightarrow \neg f \equiv 1 - f(x) \in L(X)$;
4. Если последовательность функций $f_1, ..., f_n, ... \in L(X)$, то $\overset{\infty}{\underset{n=1}{+}} f_n(x) \in L(X)$, $\overset{\infty}{\underset{n=1}{\bullet}} f_n(x) \in L(X)$.

Для определения пространства возможностей необходимо ввести определение меры возможностей.

Определение 2. Функцию $P: \mathrm{A} \to L$ будем называть мерой возможности, если:

1. $P(A) \ge 0$ для $\forall A \in \mathrm{A}$;
2. $P(A)$ счетно-аддитивная, т.е., для $\forall \{A_i\}_{i=1}^{\infty}, A_i \in \mathrm{A}$ такой, что $\bigcup_{i=1}^{\infty} A_i \in \mathrm{A}$ выполняется:

$$P(\bigcup_{i=1}^{\infty} A_i) = \overset{\infty}{\underset{i=1}{+}} P(A_i) = \sup_{i=\overline{1,\infty}} P(A_i).$$

Отметим, что, в отличие от классической теории меры, условие счетной аддитивности не накладывает ограничений на множества $\{A_i\}_{i=1}^{\infty}, A_i \in \mathrm{A}$ — в теории возможностей не является необходимым условие

$A_i \cap A_j = \varnothing, i \neq j$. Отсутствие этого условия в теории возможностей не является существенным дополнением или отличием от классической теории меры. Можно показать, что свойство 2, которое введено только для множеств, которые не пересекаются, может быть легко расширено на любые множества.

Выделим простейшие свойства меры возможности.

Лемма 1. Если $A, B \in \mathsf{A}$, $A \subseteq B$, то $P(A) \leq P(B)$.

Лемма 2 (непрерывность относительно монотонно возрастающей последовательности).

Пусть задано $\{A_i\}_{i=1}^{\infty}, A_i \in \mathsf{A}$ и $A_i \subset A_{i+1}$, причем $\bigcup_{i=1}^{\infty} A_i \in \mathrm{A}$. Тогда $P(\lim_{n\to\infty} A_n) = \lim_{n\to\infty} P(A_n)$.

Лемма 3 (полунепрерывность снизу относительно монотонно убывающей последовательности).

Пусть задано $\{A_i\}_{i=1}^{\infty}, A_i \in \mathsf{A}$, $A_{i+1} \subset A_i$, и $\bigcap_{i=1}^{\infty} A_i \in \mathsf{A}$, тогда $P(\lim_{n\to\infty} A_n) \leq \liminf_{n\to\infty} P(A_n)$.

Свойство полунепрерывности возможности означает, что для произвольной последовательности $\{A_i\}_{i=1}^{\infty}, A_i \in \mathrm{A}$, такой, что $\lim_{n\to\infty} A_n = A$, мы не можем в общем случае говорить, что $\lim_{i\to\infty} P(A_i) = P(A)$. С практической точки зрения для убывающей последовательности множеств мы не можем определить возможность появления предельного множества, имея возможности элементов последовательности. А именно, например, невозможно определить значение $P(\varnothing)$ по непрерывности, поскольку $P(A)$ не непрерывная при $A = \varnothing$.

Пусть X — произвольное пространство, A — определенная на пространстве X алгебра множеств, а $P(\cdot)$ - возможность на A. Естественном есть вопросы продолжения меры на более широкий класс множеств. Для этого введем понятие внешней меры.

Мера, определенная на алгебре позволяет характеризовать только узкое множество событий. Естественно возникает желание построить модель пространства событий, позволяющее изучать и описывать любое событие. Тем самым возникает вопрос о продолжении меры на множество всех подмножеств.

Определение 3. Функцию $P^*(\cdot): \beta(\mathrm{X}) \to L$, которая задается как $P^*(B) = \inf_{\{E_j\} \in \mathbf{A}} \sup_j P(E_j)$, где $\{E_i\}_{i=1}^{\infty}, E_i \in \mathsf{A}$ такие, что $B \subset \bigcup_{j=1}^{\infty} E_j$, будем называть внешней мерой возможности.

Хотя бы одно такое покрытие всегда существует. Действительно, положив $E_1 = \mathrm{X}$, $E_2 = E_3 = \ldots = \varnothing$, получим что любое множество A из X покрываются $\bigcup_{j=1}^{\infty} E_j = \mathrm{X}$.

Лемма 4. Для произвольного множества $A \in \mathsf{A}$ внешняя мера возможности равняется мере возможности этого множества, т.е. $P^*(A) = P(A)$.

Лемма 5. Внешняя возможность является неотрицательной функцией множеств, т.е. $P^*(A) \geq 0$ для $\forall A \subset \mathrm{X}$.

Лемма 6. Внешняя мера возможности $P^*(\cdot)$ монотонна, т.е. для $\forall A, B \subset \mathrm{X}$ таких, что $A \subset B$, будем иметь $P^*(A) \leq P^*(B)$.

Теорема 1 (о продолжении меры возможности). Внешняя мера P^*, которая определена на булеане $\beta(\mathrm{X})$, является мерой возможности.

Продолжая меру, мы получаем лишь верхнюю оценку, которая формально удовлетворяет свойствам меры. Заметим, что формулировка и доказательство этой теоремы существенно отличается от подобной предложенной в [11].

Свойства возможности ясно выражают мысль, что введение лишь этого варианта меры в пространстве событий недостаточно для адекватного и качественного описания моделей. Итак, существует потребность введения нижней оценки меры, физическая суть которой в том, что некоторое событие должен состояться.

Причем, мера того, что этот факт состоится наверняка, не может быть меньшей некоторого числа. И вполне естественно, что эта новая мера может быть введена на той же шкале $L=([0,1],\leq,+,\circ)$, что и возможность, если мы стараемся описать событие двумя величинами, которые должными быть каким-то чином связанные.

Определение 4. Мерой необходимости будем называть функцию $N:\mathrm{A}\to L$, которая удовлетворяет таким условиям:

1. $N(A)\geq 0$ для $\forall A\in\mathrm{A}$;
2. $N(A)$ — счётно-мультипликативная, т.е., для $\forall\{A_i\}_{i=1}^{\infty}, A_i\in\mathrm{A}$ такой, что $\bigcap\limits_{i=1}^{\infty}A_i\in\mathrm{A}$, выполняется:

$$N\left(\bigcap_{i=1}^{\infty}A_i\right)=\mathop{\circ}_{i=1}^{\infty}A_i=\inf_{i=1,\infty}N(A_i).$$

Суть условий, которые накладываются на функцию $N(\cdot)$, можно интерпретировать следующим образом: для двух событий, которые состоятся с некоторыми значениями необходимостей, совместное событие состоится с наименьшей (минимальной) необходимостью.

Мера необходимости обладает свойствами монотонности, непрерывности относительно монотонно убывающей последовательности и полунепрерывности относительно монотонно возрастающей последовательности.

Меры возможности и необходимости обладают дуальными свойствами. Таким образом, их логично рассматривать в паре, поскольку вместе их преимущества дополняют друг друга, что дает позволяет описывать более точно не только события, но и операции над ними.

Итак, теперь будем рассматривать $(\mathrm{X},\mathrm{A},P,N)$ модель пространства состояний с двумя мерами, которую будем называть (PN)-моделью, считая ее адекватной для описания поставленных в начале работы вопросов.

Введя в построение модели пространства состояний понятие возможности и необходимости, мы получили аппарат, который позволяет оценивать сверху и снизу события.

Если мы используем в модели пространства событий две меры, то естественным образом возникает вопрос в продолжении меры необходимости, как это было сделано для меры возможности.

Определение 5. Функцию $N_*(\cdot):\beta(\mathrm{X})\to L$, которая задается как $N_*(A)=\sup\limits_{\{E_j\}}\inf\limits_{j=1,\infty}N(E_j)$, где $\{E_i\}_{i=1}^{\infty}, E_i\in\mathrm{A}$ такие, что $A\supset\bigcap\limits_{j=1}^{\infty}E_j$, будем называть внутренней мерой необходимости.

Теорема 2 (о продолжении необходимости). Пусть задана модель пространства состояний $(X, \mathbf{A}, P, N)$. Меру необходимости N можно продолжить с алгебры A на булеан $\beta(\mathrm{X})$ с сохранением свойств счетной мультипликативности и неотрицательности.

Таким образом, для модели $\{\mathrm{X}, \mathsf{A}, P, N\}$ мы получаем обе оценки P^* и N_* для описания любого события из X.

Персептивные элементы

Персептивность возникает в тот момент, когда появляется потребность описывать то или другое событие с точки зрения субъективного восприятия действительности. Пусть X — пространство реальных объектов, которые мы исследуем. A — некоторая заранее заданная алгебра событий, которые могут состояться, а P и N — соответственно мера возможности и необходимости, которые определены для событий алгебры A. В дальнейшем будем рассматривать (PN)-модель $\{\mathrm{X}, \mathsf{A}, P, N\}$ пространства возможностей. Введем также пространство (Y, B) элементов, которые являются отображением действительности.

Определение 6. A, B-измеримую функцию $\xi: \mathrm{X} \to \mathrm{Y}$, заданную на $(\mathrm{X}, \mathrm{A}, P, N)$, которая принимает значение в Y, будем называть *персептивным элементом*.

Это определение говорит о том, что для конкретного ситуативного восприятия действительности (обозначенного буквой ξ) нашим внутренним величинам из Y отвечает какое-либо определенное состояние (или даже не одно) реальности $\xi^{-1}(y) \subset \mathrm{X}$.

Назовем функцию $\varphi_\xi(y): \mathrm{X} \to L$, такую что задается равенством $\varphi_\xi(y) = P\{x \mid \xi(x) = y\}$, распределением возможностей персептивного элемента ξ. А функцию $\psi_\xi(y): \mathrm{X} \to L$, которая задана как $\psi_\xi(y) = N\{x \mid \xi(x) = y\}$, распределением необходимости персептивного элемента ξ.

Определение 7. Функцию $\eta: \mathrm{X} \to \beta(\mathrm{Y})$ будем называть η*-персептивным множеством*.

Приведем также альтернативное определение персептивного множества.

Определение 8. Совокупность персептивных элементов $A = \{\xi_j\}, j \in J$ будем называть A-*персептивным множеством*.

Установим взаимосвязь между этими определениями.

Теорема 3 (об эквивалентности определений**).** η-персептивное множество можно представить в виде определения A-персептивного множества и наоборот.

Замечание. Как видно из доказательства, мы можем неоднозначным образом представить η-персептивное множество в виде A-персептивного множества.

Согласно классической теории множеств, мы можем определить понятие пустого персептивного элемента и пустого персептивного множества, а именно следующим образом: $\xi_0: \mathrm{X} \to e$; $\eta_0: \mathrm{X} \to \varnothing$;

Легко убедиться, что понятие персептивного элемента ничем не отличается от понятия нечеткого элемента, введенного в работе Пытьева Ю. А. [11]. Но в отличие от понятия нечеткого множества, введенного в указанной работе, понятие персептивного множества существенно отличается.

Для построенных множеств вводятся основные операции: пересечение, объединение, разность, дополнение, включение.

Замечание.

- пустое множество будем рассматривать как множество $\eta_\varnothing(x)$ такое, что $\forall x \in \mathrm{X} \;\; \eta(x) = \varnothing$;

- под универсальным персептивным множеством будем понимать множество η_{univ} такое, что $\forall x \in \mathrm{X}$ $\eta_{univ}(x) = \mathrm{Y}$;

- пространство всех η-персептивных множеств будем обозначать через $\beta(X,Y)$.

Для операций, введенных таким образом, нетрудно убедиться, что дополнение $\overline{\eta}$ можно представить в виде $\overline{\eta} = \eta_{univ} \setminus \eta$.

Также легко убедиться, что выполняются следующие элементарные свойства: идемпотенции, коммутативности, ассоциативности, абсорбции, дистрибутивности, комплиментарности. Это вытекает непосредственно из того, что указанные законы выполняются для каждого фиксированного $x \in \mathrm{X}$ (поскольку для $\forall x \in X\ A(x), B(x), C(x) \subset \mathrm{Y}$ эти свойства выполняются).

Персептивный элемент порождает пространство возможностей $(\mathrm{Z}, \mathbf{B}, P_\xi)$, где возможность для каждого $B \in \mathbf{B}$ задаётся как $P_\xi(B) = P(\xi^{-1}(B)) = P\{\xi(x) \in B\}$.

Из теоремы 1 следует, что можно определить функцию $\mu_\xi(y) = \overline{P}\{\xi = y\}$, которая называется распределением персептивной величины. Возможность того, что персептивная величина ξ попадёт в множество *B*, можно выразить через распределение: $P\{\xi \in B\} = \sup_{y \in B} \mu_\xi(y)$.

Вернемся к эксперименту с измерением температуры воды. Пусть есть некий прибор для измерения температуры. Известно, что точно можно измерить температуру $0^0 C$ и $100^0 C$ при атмосферном давлении 760мм рт. ст. Все остальные значения отображаемы прибором будут неточными. Из естественных соображений следует, что универсальное пространство $X = [0; 100]$. Пусть $Y = [-5; 110]$ — множество значений температуры, которые может выдать прибор с учетом (большим запасом) погрешностей. Также пусть экспертами построена персептивная величина $f(x)$, описывающая работу прибора такая, что $f(0) = 0$ и $f(100) = 100$. Тогда распределение возможностей значений персептивной величины можно описать функцией $\mu_\xi(y) = P\{\xi = y\} = \max\{0, 1 - |y - \xi^{-1}(y)|\}$. Тем самым мы можем построить комплексную модель эксперимента.

Нечеткая динамика

Часто при решении прикладных задач, используются математические модели, в которых в качестве параметров выступают неопределенные нечеткие величины. Например, пусть необходимо вычислить траекторию движения судна при ветреной погоде. Причем скорость ветра характеризуется понятиями: слабый, нормальный, сильный, ураган и т.д. Либо скорость измеряется прибором с известным распределением возможностей значений. Тогда для описания динамики предлагается использовать дифференциальное уравнение вида

$$y(t,x) = y_0(x) + \int_{t_0}^{t} a(y(s),s)ds + \int_{t_0}^{t} b(y(s),s)dw(s,x)\ ,$$

где последний интеграл является интегралом по процессу нечеткого блуждания. Его построение основывается на предложенной выше теории.

Для этого дифференциального уравнения доказана теорема о существовании и единственности решения, предлагаются новые определения устойчивости решений и доказаны соответствующие теоремы.

Заключение

Таким образом, в рамках теории возможностей получена модель для описания субъективного представления. Данная модель дает возможность не только моделировать неточно полученные величины, лингвистические выражения, различные неопределенности, но и построить комплексную модель эксперимента, включая количественную характеристику неопределенности. Такой характеристикой являются меры возможности и необходимости.

С помощью полученных понятий можно, например, определить естественный интеллект, как функциональное отображение $\xi : X \to X$, сообщество носителей интеллекта, как отображением $\eta : X \to X$, где X — универсальное множество. Или, другими словами, естественным интеллектом можно назвать свойство (например, человека) A элемента множества $B \subset X$ (например, общества), приобретенное в процессе взаимодействия всех элементов универсального множества X.

Тогда искусственный интеллект следует рассматривать как свойство некоторого элемента D универсального множества X, не являющегося элементом B, приобретенное в процессе взаимодействия конечного количества элементов универсального множества.

ЛИТЕРАТУРА

[1] Бехтерев В.М. Основы учения о функциях мозга. С.-Пб. Изд. Брокгауз и Ефрон, 1907., 512 стр.

[2] Айзенк Г.Ю. Интеллект: новый взгляд // Вопр. психологии, №1, 1995, с. 111 – 131.

[3] Guilford Y. P. The nature of human intelligence. - N. Y.: Mc-Graw Hill, 1967.

[4] Шопенгауэр А. Мир как воля и представление, Т.2 / Пер. с нем. -Мн.: ООО "Попурри", 1999. - 832 с.

[5] Zadeh L.A. Fuzzy sets.- Information and Control.- 1965.- 8, 3, 338-353.

[6] Zadeh L.A. Fuzzy sets as a basis for a theory of possibility. - Fuzzy Sets and Systems, 1, 1978, 3 - 28.

[7] Sugeno M. Theory of Fuzzy integral and It Applications // Ph.D Thesis, Tokyo Inst of Technology, Japan, 1974.

[8] Zadeh L.A Probability measures of fuzzy events // J. Math. Anal. And Appl., 23, 1968., p. 421-427

[9] Dubois D., Prade H. Fuzzy Sets and Systems: Theory and Applications // vol. 144 in Mathematics in Sciences and Engineering Series Academic Press, New York, 1980.

[10] Заде Л. А. Основы нового подхода к анализу сложных систем и процессов принятия решений.//Математика сегодня. – М.: Знание, 1974.

[11] Пытьев Ю. П.. Возможность. Элементы теории и применение. М.: УРСС, 2000. – 190с.

[12] О.Бичков Побудова інтегралу за процесом нечіткого блукання. // Вісник Київського університету, Сер.: фіз..-мат. науки, №4, 2005. – с.125-133.

[13] Чуличков А.И. Математические модели нелинейной динамики. – М.: Физматлит, 2000. – 296 с.

Информация об авторах

Alexey Bychkov - *Taras Shevchenko National University of Kyiv, Cybernetics department, ul.Volodimirska, 64, Kyjv, Ukraine, e-mail: bychkovtk@gmail.com*

Mikhail Merkuryev - *Taras Shevchenko National University of Kyiv, Cybernetics department, ul.Volodimirska, 64, Kyjv, Ukraine, e-mail: bychkovtk@gmail.com*

ПРЕДПОСЫЛКИ ВОЗНИКНОВЕНИЯ ОБЩЕЙ ТЕОРИИ ИНФОРМАЦИИ

Василий Луц

***Аннотация**: В статье рассматриваются некоторые предпосылки и факторы, обуславливающие возникновение общей теории информации, в частности, многозначность понятия информации, современные условия общества, тенденции развития науки. Изложены замечания к теории информации, описаны различные типы и формы существования информации.*

***Ключевые слова**: терия информации, обобщение, неоднозначность, базисная информация, точка зрения, метаинформация.*

***ACM Classification Keywords:** H.1.1. General systems theory, Information theory, Value of information.*

Введение

"Мы полностью захвачены своим частным взглядом на мир,
и это заставляет нас не только чувствовать,
*но и действовать так, как если бы мы знали о мире **все**."*

дон Хуан Матус

Появление и развитие теории информации вызвало значительный интерес ученых самых различных специальностей, которые имели отношение к понятию «информация» или просто использовали свойства информации в неявной форме. Но для подавляющего большинства исследователей интерес объяснялся терминологическим недоразумением, поскольку теории, описывающей свойства информации в широком смысле этого слова, пока что нет. Существующая теория связана с кодированием информации в целях ее передачи по каналам связи. Кроме техники связи, она находит еще существенные приложения в теории вычислительных машин [Фано, 1965].

Вышесказанное еще раз подтверждает, что информация – это одно из наиболее общих понятий и стоит в одном ряду с такими базисными, основополагающими категориями, как материя, энергия, время, пространство. При этом информация неразрывно связана с ними разнообразными взаимоотношениями, описание, представление которых может значительно различаться в различных научных (или ненаучных) школах и течениях. Их можно сгруппировать в две большие группы: одни ученые считают, что информация – это отражение существующих в пространстве и времени материальных изменений, существующее и проявляющееся в различных материальных носителях, т.е. информация – неотделимое свойство материи, не существующая сама по себе. Другая группа рассматривает альтернативный вариант, в рамках которого информация – независимая от других сущность, имеющая собственные свойства и проявления. Но, независимо от точки зрения различных исследователей, представляется бесспорным следующий факт: понятие информации требует детального изучения, что в конечном результате приведет к возникновению *общей теории информации* (ОТИ), имеющей разнообразные приложения практически во всех областях человеческого знания. Целью данной статьи является рассмотрение некоторых факторов и предпосылок, обуславливающих возникновение и развитие ОТИ.

Многозначность понятия информации

Интерес к понятию информации, стремление понять ее природу, а также повсеместное использование разнообразных свойств информации позволяет сделать предположение, что за данным термином может скрываться несколько различных, хотя и взаимосвязанных, представлений. Другими словами, смысл, вкладываемый в различных ситуациях в одно и то же понятие «информация», может значительно отличаться. При этом возможны два варианта: либо конкретная проекция многозначного термина

определяется в процессе коммуникации, т.е. зависит от контекста, либо возникает некоторая степень непонимания, искажения при передаче информации (если в каждом конкретном случае провести детальный анализ, о чем же именно говорится, то это, возможно, дополнит наши представления об информации вообще). В принципе возможно и сочетание этих вариантов, когда смысл слова в какой-то мере определяется из контекста, но все равно возникает некоторое недопонимание (подобными примерами жизнь нас снабжает регулярно, возможно, даже сейчас, когда вы прочитаете эту статью). Понятие информации не является исключением из правил, множество слов естественного языка отличаются многозначностью использования, что и порождает существенные трудности для полноценного машинного перевода. Но если большинство многозначных слов получили достаточно детальное описание их различных толкований, то понятие информации еще ждет своих исследователей.

Для наглядности изложения можно отметить некоторые формы информации, которые в той или иной мере различаются между собой:

- непосредственная – опосредованная;
- первичная – вторичная (в т.ч. и метаинформация);
- потенциальная – существующая;
- абстрактная – конкретная;
- выраженная – неявная;
- структурированная – слабоструктурированная;
- виртуальная – реальная;
- внутренняя – наблюдаемая;
- активная – пассивная;
- основная – косвенная;
- основная – дополнительная;
- согласованная – разрозненная;
- однородная – разнородная.

Предлагаемая классификация достаточно условна и может быть в перспективе выражена более подходящими терминами, не вызывает сомнений сам факт существования разных типов и форм информации.

Непосредственная информация возникает и существует только в некий определенный момент, порождая различные формы опосредованной информации в процессе всевозможных воздействий на окружающую среду. Опосредованная информация, являясь отражением непосредственной, может существовать в форме различных материальных изменений (материальных носителей информации), а также в виде полей. Принципиальное различие между непосредственной и опосредованной информацией заключаются в том, что непосредственная является полной (содержащей все существующие нюансы) и взаимосогласованной, а опосредованная – только отпечатком, проекцией непосредственной информации, всегда появляющейся на многовение позже, и по определению является в той или иной мере неполной, неточной, искаженной (уже в процессе своего возникновения), а также не всегда взаимосогласованной между собой. В процессе изменений непосредственная информация о моменте времени t_i исчезает, накладывая некоторый отпечаток на возникающую в следующий момент времени t_{i+1} непосредственную информацию (и соответствующие материальные изменения). Здесь возникает вопрос: а как же соотносится непосредственная информация о моментах времени t_{i+1} и t_i, насколько полно передается внутренняя информация во времени? Вероятно, это зависит от дополнительных факторов, но в общем случае можно отметить, что существует принципиальная неопределенность (в частности, неточность передачи информации, как в пространстве, так и во времени), что в какой-то мере иллюстрируется неопределенностью Гейзенберга. Аналогично, опосредованная информация искажается, теряется, т.е. под действием различных разрушающих факторов видоизменяются ее материальные носители (сверхслабые "торсионные" поля также можно считать материальными носителями, даже если они не всегда регистрируются). Разрушающие факторы приводят не только к потере существовавшей

информации, но и вносят другую информацию – как результат воздействия других факторов, порождающих информацию, а также информацию о произошедших изменениях (в том числе – о проявлении чисто деструктивных сил, под влиянием которых происходит нарушение структур, ослабление, искажение, разрушение связей).

Существует, правда, еще и другая возможность для сторонников полевой концепции, а именно: возникающие поля не подвергаются затем никаким разрушительным воздействиям, таким образом, навсегда сохраняя информацию обо всех происходящих изменениях. Последнее замечание о сохранении полной информации о прошедшем событии можно представить и в другой форме, с другой позиции: сохраняются не сами поля, а метаинформация о них (опять же в виде некоторых вторичных полей, отображающих первичные), но в этом случае происходит своего рода обобщение и соответственно сжатие непосредственной информации (с потерями или без).

Первичная информация включает в себя как непосредственную, так опосредованную первого уровня, а вторичная – различные формы отображения опосредованной информации, в частности, мета-информацию, информацию об информации, и просто преобразования опосредованной информации в другую форму (при этом видоизменяется форма, а в сущности передается все та же информация). Очевидно, что в процессе дальнейших преобразований вносятся те или иные изменения, вызываемые характером воздействия и свойствами материальных носителей (в частности, они могут носить как положительный, уточняющий характер, так и деструктивный – внесение шума, ненужных данных, искажение первичной информации). В общем случае форма выражения, представления информации накладывает определенный отпечаток, вносит дополнительную информацию и некоторые искажения. Простая иллюстрация – существуют различия (и порой существенные) между тем, что человек понимает (или даже просто хочет сказать), и тем, что именно он сказал (влияние выбранных форм языковых конструкций), а также непосредственно той информацией, которую смог извлечь собеседник (в условиях ограничений по времени, наличия собственного мнения и т.д.). В этом случае достаточно хорошо заметны основные особенности преобразования информации из одной системы координат в другую – изменение избирательности и масштаба отображения. Сочетание определенного масштаба и избирательности (выбора базисных положений, а также рассматриваемых объектов, параметров, свойств) можно назвать ракурсом отображения. При этом для различных элементов отображаемого может применяться тот или иной масштаб (имеется в виду параллельное использование различных масштабов), что обуславливает множество возможных ракурсов отображения.

Потенциальная информация представляет собой отображение всевозможных изменений, которые согласуются с существующими в данный момент. Возникает вопрос: а где, в чем содержится эта потенциальная информация? Видимо, она содержится в неявной форме в соотношениях существующих особенностей, проявлений как возможность их реализации в будущем. При этом реализуется лишь малая часть того, что может быть (во-первых, по причине наличия альтернатив – если человек сидит, он не может одновременно бежать, плыть и т.п., во-вторых, по причине существования различных ограничений – некоторые возможности не реализуются, если им не хватает энергии для реализации или преодоления неких внешних препятствий). Вообще говоря, потенциальная информация намного больше по объёму, чем существующая, что создает определенные трудности при её оценке. Кроме этого, потенциальная информация может базироваться на тех изменениях, которые не отображаются в используемых системах координат. В общем случае для её рассмотрения необходимо использовать методы кластерного анализа, т.е. оценивать возможные пересечения, объединения по различным критериям.

Информация, выраженная в определенной форме, использует, включает неявную информацию, которая может быть двух типов: во-первых, подразумеваемая информация (в частности, базисная информация, необходимая для понимания высказанной, передаваемой информации); во-вторых, информация, непосредственно вытекающая из существующей (путем применения некоторых правил преобразования и согласования).

Структурированность информации вытекает из ее природы как отображения различий, и подразумевает ее однозначное представление в форме некоторых структур с выбранных позиций. Слабоструктурированной (или не структурированной) информация является в случае существования широкого диапазона возможных значений рассматриваемых свойств, характеристик, базовых положений, а также неоднозначности их комбинирования, что порождает множество различных интерпретаций и проекций. Структурированность информации подразумевает наличие связей между ее элементами, следовательно, когда связи и отношения выражены в явной форме, информация является структурированной, а когда в неявной (или потенциальной) – слабоструктурированной. Отметим, что связи бывают вертикальные (иерархические), горизонтальные, одного уровня детализации (в том числе – синонимы и антонимы), и мета – связи (возникающие при неоднократном отображении первоначальной информации).

Информация называется реальной, если она отображает нечто реально существующее. Виртуальная же информация получается в результате комбинирования отдельных не связанных между собой информационных структур в единое целое, не имеющее прообраза в материальной форме (например, сфинкс, кентавр, русалка и т.д.), часто виртуальность понимают как существование только на уровне информации, без материального воплощения (виртуальные миры компьютерных игр, виртуальная лаборатория или университет и т.д.), хотя это не совсем точно – на самом деле виртуальная лаборатория имеет определенные формы материальных носителей.

Внутренняя информация определяет изменение в следующий момент времени (и отображает предыдущие изменения), а внешняя появляется как результат отображения этих изменений в окружающей среде. Это различие, видимо, имел в виду Кант, когда писал о "вещи в себе" и "вещи для других". Очевидно, что отображение изменений в большой мере определяется не только характером изменений, но и характеристиками внешней среды (которые проявляются в форме различных материальных носителей). Таким образом, внешняя информация носит неполный и неточный характер, поскольку могут существовать иные критерии сравнения, другие объекты, которые могут отобразить рассматриваемые изменения с другой стороны, т.е. другие их качества (процесс взаимодействия – тоже сравнение, поскольку результат определяется внутренними свойствами взаимодействующих объектов).

Основная информация определяет основные, наиболее весомые элементы и соотношения, а косвенная – дополнительные, слабо связанные с основными (например, ассоциативными, ситуативными связями). Соответственно, основная информация предопределяет дополнительную (существующую как в неявной форме, так и в явной, в последнем случае ее можно иногда удалить, а затем снова получить, исходя из основной информации, базисных положений и причинно-следственных связей).

Детальное рассмотрение всех типов информации (точнее, всех вышеперечисленных ее типов), а также взаимосвязей между ними выходит за рамки данной статьи.

Современные условия существования общества

Последнее время в статьях и книгах самых разных направлений встречаются той или иной форме мысли, соображения об особенностях современного мира, среди которых ведущую роль занимают, во-первых, увеличение объемов информации (явление информационного взрыва), во-вторых, ускорение различных информационных процессов (иногда говорят про уплотнение времени, т.е. увеличение количества событий в единицу времени). Соответственно, выделяют некоторые последствия этих явлений, как положительные, так и деструктивные. Среди первых основное место занимает растущее количество возможностей, которые предоставляет современное положение вещей (действительно, по сравнению с людьми прошедших эпох мы имеем намного больше самых разнообразных возможностей). Но количество возможностей порождает задачу, (а точнее, проблему) их согласования в условиях различного рода ограничений – ограниченности времени, ресурсов, финансов, энергии, наличия стратегических

приоритетов и т.д. Сложность последней привело к возникновению такого течения, как "slow life", имеющего разнообразные проявления ("slow food", "slow time" и др.)

В общем случае ограниченность ресурсов и наличие множества возможностей разного рода (разнообразие форм досуга, возможности коммуникации, доступ к огромному количеству разнородной, но интересующей информации, и т.д.) приводит к тому, что человек на практике достигает меньших результатов, чем мог бы (в частности, при условии полной концентрации своих усилий в одном направлении). Соответственно, существует необходимость согласования деятельности людей, занимающихся в одной сфере, с целью достижения максимально возможного результата (или хотя бы уменьшения потерь при передаче важной для исследований информации).

Можно отметить, что дальнейшее развитие цивилизации подразумевает естественное развитие общей теории информации. Этому способствуют такие факторы, как развитие дистанционного образования, самообразования, коммуникации, повышение эффективности обучения. По любой теме можно найти множество различных материалов разного уровня сложности, степени детализации, использующих различные формы представления информации. Следовательно, рано или поздно процессы согласования, изложения, анализа с различных позиций имеющейся информации приведут к качественно новому подходу, основанному на понимании относительности, ограниченности любой известной информации, изначально рассматривающему существование альтернативных точек зрения. Ускорение рассмотренных процессов возможно, хотя и требует выполнения дополнительных условий различной степени сложности. Видимо, существует тип исследователей, которым от природы легко работать с неопределенностью, наличием альтернатив, различных потенциальных возможностей, большими объемами новой информации различных типов, и необходимо объединение их усилий для достижения поставленных целей. В то же время некоторым ученым, среди которых могут быть и профессионалы высокого уровня, будет достаточно сложно принять и использовать ОТИ, как вследствие психологических особенностей восприятия и обработки информации, так и под влиянием инерции полученного образования, определенным образом организованного жизненного опыта.

Логика развития науки

История развития науки демонстрирует постоянное увеличение уровня абстракции (идеал научной теории обрисован еще И. Ньютоном: “Объяснить как можно большее количество фактов как можно меньшим числом исходных положений”). Увеличение количества ученых, базирование на достижениях предыдущих поколений позволяют добиться значительных успехов, но периодически возникает необходимость в значительном перепросмотре, переоценке имеющихся теорий, при этом либо возникает новый взгляд на известные вещи, либо происходит коррекция, расширение точки зрения под влиянием ранее неизвестной информации.

Достижения различных существующих наук так или иначе используют свойства информации, в достаточно яркой форме это проявляется в психологии, философии. Например, в психологии (НЛП) наглядно демонстрируются преобразования базисных положений точек зрения: рассмотрение первопричины в другой форме проявления – императив позитивных намерений и переопределение, выбор масштаба оценки поступков – изменение размеров фрейма, нахождение положительных сторон – рефрейминг. В философии рассмотрены некоторые важные моменты, например, Л. Витгенштейн в своем "Логико-философском трактате" утверждает, что для того, чтобы что-то рассмотреть, оценить, обязательно выбрать некоторые базисные положения, которые принимаются без всякого сомнения, без которых невозможно что-либо сказать (они как бы выполняют роль неподвижных дверных петель, пишет он, позволяя варьировать, изменять, оценивать другие факты).

Определенную роль может сыграть и развитие теории торсионных полей (Г.И. Шипов и др.), где предполагается, что информация (как непосредственная, так и опосредованная) возникает и существует в форме полевых образований, которые могут взаимодействовать друг с другом и с материальными

объектами. Хотя оценки данной теории носят неоднозначный характер, но, исходя из конструктивного подхода, из любой теории можно извлечь некоторую информацию (полезную).

В общем случае дальнейшее развитие информационных технологий непосредственно связано с более глубоким пониманием сущности информации [Луц, 2006]. Развитие поисковых систем, расширение служебной информации, организация удобного доступа, надежного хранения и защиты, эффективной обработки информации и др. – практически всё требует значительного пересмотра существующего положения вещей. Одно только отделение и рассмотрение базисной информации, которая определяет форму и особенности представления данных, может сыграть значительную роль в упорядочении накопившегося огромного информационного массива.

Развитие систем искусственного интеллекта

Если использовать существующие методы представления и обработки информации, многие задачи практически не подлежат решению. Даже построение полноценной компьютерной модели жизнедеятельности простейшей формы жизни – кишечной палочки *Escherichia coli* (одного из наиболее изученных видов микробов) наталкивается на непреодолимые технические трудности (реализованы лишь отдельные проекции – модели движения, питания, да и то частично), хотя для этого исследования создан международный альянс (International E. coli Alliance — IECA), распределивший общую задачу между множеством лабораторий. Чтобы виртуальная модель бактерии могла плавать, питаться, отбиваться от вирусов, копировать свою ДНК и выполнять массу других обычных для нее задач в одно и то же время, необходимо 5-10 лет работы ученых (и это с учетом тенденций развития компьютерной техники).

Возникает вопрос: а не существуют ли альтернативные подходы моделирования? Автор считает, что возможна разработка необходимого теоретического аппарата, хотя для этого и понадобятся определенные усилия (и соответствующие затраты). В первую очередь необходимо создать специализированные средства для представления, преобразования, согласования информации, позволяющие осуществлять эффективное взаимодействие исследователей, а также усовершенствовать условия хранения и классификации различной информации (в частности, для удобства доступа к источнику информации, поскольку достоверность получаемой информации является одной из важнейших характеристик). В перспективе расстояние между получением новой достоверной информации и ее внедрением может значительно сократиться, хотя в данный момент наблюдается противоположное.

В общем же случае для разработки различных систем искусственного интеллекта необходимо провести значительную предварительную работу по анализу используемых человеком очевидных, подразумеваемых представлений (и переводе их в явную форму). Результаты такой работы можно будет использовать также и в системах поддержки коммуникации ученых, во-первых, для отображения возможных вариантов, возможностей интерпретации, во-вторых, для согласования информации. Без значительного развития систем искусственного интеллекта практически невозможно будет добиться эффективной обработки, согласования, использования разрозненной, разнородной информации.

Замечания к теории информации

Рассматривая теорию информации, можно отметить следующее. Во-первых, существует известная (как отправителю, так и адресату), используемая информация и передаваемая (которая передается адресату именно по причине ее неизвестности), и это различие играет существенную роль. В теории информации акцент был сделан на передаваемой информации, поскольку основной целью являлась её передача по каналам связи, и данное различие не рассматривалось. Известная информация, используемая по умолчанию, состоит из базовой (первоначальной) и согласующейся с ней получаемой (возникающей) информации. Существование базовой информации и возможности её выбора, варьирования (и как следствие – наличие альтернативных подходов) также не рассматривалось в теории информации. Исходя из вышеизложенного, можно, увеличивая количество известной, базовой информации, уменьшать объем передаваемой информации.

Во-вторых, информация понимается как однозначное отображение, и не рассматривается ситуации, когда альтернативные критерии различия, соответствующие другой точке зрения, порождают иную информацию. Представим, что мы наблюдаем систему, которая может принимать N возможных состояний. Тогда, согласно Хартли, информация о системе равна $\log N$. Нетрудно представить существование другого критерия, согласно которому количество возможных состояний будет меньше или больше N, и соответственно, будет другая информация. Изменение N возможно при увеличении или уменьшении разрешающей способности наблюдателя, а также в случае динамических изменений (слияние некоторых состояний или дифференциация одного из наблюдавшихся на два и более), не говоря уже про альтернативные точки зрения, для которых критерии значимости и различимости состояний могут быть самыми разными. Можно возразить: независимо от внешнего наблюдателя существует объективное количество состояний системы, которому соответствует вполне однозначная внутренняя информация, конкретная внутренняя определенность. Но отображение этой внутренней информации принципиально неоднозначно, а именно отображение, образ системы используется различными внешними воспринимающими объектами и субъектами в качестве основы для взаимодействия с системой. Таким образом, можно выделить как минимум две проекции информации: внутреннюю определенность и внешнее отображение этой определенности с различных позиций.

Что же представляет собой внутренняя определенность? Очевидно, что она предопределяет изменение в последующий момент времени (и в то же время является следствием определенности, существовавшей в предыдущий момент). Другими словами, определенность в данный момент времени отображает определенность, существовавшую в предыдущий момент времени, т.е. тоже является отображением, но не в пространстве, а во времени, и на это отображение также действуют различные факторы, вносящие искажения. Внутренняя определенность проявляется как повторяемость, пролонгированность во времени, закономерность изменений. Для лучшего понимания можно представить гипотетический вариант, когда все изменения носят только непредсказуемый, случайный характер (хотя подобное вообще-то трудно представить в полной мере). В последнем случае (ситуации полной неопределенности) будет действовать только один фактор, который противоположен определенности. Под информацией обычно подразумевают именно определенность, хотя в общем случае отображение может включать также различные формы неопределенности.

Могут существовать различные, как альтернативные, так и дополняющие друг друга, подходы и модели для описания информационных процессов. Рассмотрим некоторые обобщения модели системы связи на случай возникновения и отображения изменений. Самая простая модель имеет следующий вид:

При этом источником изменений может быть любой источник, генерирующий информацию, а каналами связи, передающими возникающие изменения (а также определяющие характер и особенности отображения изменений) могут быть любые существующие среды, поля и формы взаимодействия. Внешняя среда, получающая информацию, может состоять из различного рода элементов и структур, в той или иной мере отображающих происходящие изменения. Рассмотрим еще модель, представленна на рис.1.

Данная блок-схема во многом напоминает используемую Р.Фано, хотя смысл, вкладываемый в используемые понятия (источника, кодеров источника и канала, декодеров канала и приемника), отличается. Очевидно, что результат кодирования (отображения изменений) передается по каналу, а в результате декодирования происходит извлечение информации, построение некоторого отображения первичного источника изменений.

В процессе передачи по каналу возможны различные преобразования информации, и в общем случае можно рассматривать последовательность каналов, связанных между собой промежуточными объектами (создающих преобразования информации из одной формы в другую).

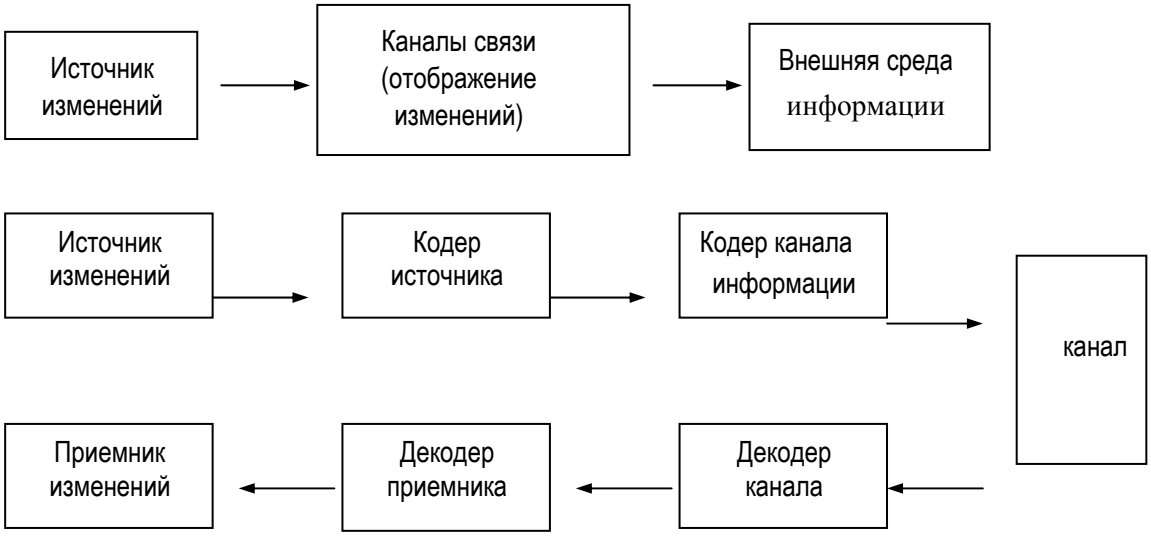


Рис. 1

Некоторые особенности общей теории информации

В заключение рассмотрим некоторые положения, которые могут быть в той или иной форме использованы при разработке общей теории информации (хотя и вытекающие из авторского понимания сущности информации).

Информация отображает существующие (или возможные) различия, но, чтобы оценить, увидеть те или иные различия (или наоборот, их отсутствие), необходимо провести операцию сравнения. И в зависимости от того, что с чем сравнивается, и будет получен результат. Операция сравнения может происходить как с позиции наблюдателя, так и в результате непосредственного взаимодействия.

Особенности возникающей информации определяются исходной, базисной информацией. Любая существующая информация – относительна и ограничена. Другими словами, существует неизвестная, непредсказуемая информация, а также просто недоступная (по причине недоступности базисной позиции, с которой возможно ее существование).

Относительность и ограниченность информации подразумевает существование альтернатив. Альтернативы могут как значительно различаться, так и быть очень близкими. Близкие альтернативы порождаются незначительными изменениями, вариациями исходной информации. Реализация одной из альтернатив не означает полного отсутствия, исчезновения других.

Существует конкуренция альтернатив: в любой момент времени пассивная (существующая в потенциальной форме) альтернатива может стать активной, вытеснив, заменив предыдущую. Факторы, вызывающие изменение альтернатив, могут быть как случайными, так и закономерными, приводящими либо к усилению пассивной альтернативы, либо к ослаблению активной.

В общем случае можно говорить о интенсивности альтернатив (как одной из причин проявлений вероятности). Характеристика интенсивности предполагает следующие варианты событий: отсутствие реализации потенциальных альтернатив при недостаточной их интенсивности; реализация одной возможности, интенсивность которой выше порогового значения; конкуренция альтернатив, имеющих необходимый для реализации уровень интенсивности (хотя реализуется лишь одна из них, либо возникает альтернатива нового типа, включающая в себя некоторые предшествующие). Факторы, усиливающие интенсивность альтернатив, могут носить как внутренний, так и внешний характер. Рассмотрение понятия интенсивности предполагает, подразумевает отдельное, в какой-то мере независимое существование альтернатив (в форме полевых образований).

Возможно сосуществование различных качеств, свойств в следующих формах: а) разделение в пространстве б) разделение во времени; а также появление нового, интегрального качества.

Существуют два варианта отображения, выделения различий: использование некоторого критерия либо просто обозначение элементов непосредственного опыта (это X, а это – Y). Второй способ является более простым и естественным, и во многом определил развитие естественного языка. При этом часто наблюдается следующее явление: "что не названо, то не существует". Другими словами, теряют, забывают различие между картой и территорией, между моделью и реальностью.

Преградами на пути разработки общей теории информации могут быть: огромное количество существующей разнородной информации; значительные затраты на ее согласование, преобразование (как теоретического характера, так и технического); сопротивление на разных уровнях некоторым положениям ОТИ (например, о принципиальной ограниченности, относительности любой информации, возможности альтернативных точек зрения).

В общем случае информация может включать в себя отображение как существующей определенности, так и различных форм неопределенности (флуктуаций, наличие диапазона возможных значений, кумулятивных факторов, влияние ситуативных связей, появление новых параметров, характеристик, и т.д.). Положительный аспект определенности заключается в существовании устойчивых структур, закономерных последовательностей изменений, а отрицательный – в сохранении неизменности, предопределенности изменений. Соответственно, положительной стороной неопределенности является принципиальная возможность внесения изменений, увеличение разнообразия (появление случайных флуктуаций, непредсказуемых изменений), а отрицательной – разрушение, ослабление существующих связей, форм (без возникновения новых).

Выводы

В статье рассматриваются некоторые предпосылки и факторы, обуславливающие возникновение и развитие общей теории информации. Существование различных форм и проявлений информации предполагают возможность (и необходимость) её более глубокого исследования, а лавинообразное накопление огромного объема разнообразной информации, ведущее к уменьшению возможностей ее полноценной оценки, использования и даже просто восприятия отдельной личностью, требует создания новых инструментов и форм для повышения эффективности коллективной работы.

Возможности научных работников качественно возросли, также значительно увеличилось их количество, что является благоприятными факторами как для проведения научных исследований, так и для подготовки высококвалифицированных кадров путем развития методов и форм обучения (самообучения), передачи опыта.

Общая теория информации предполагает построение, сравнение различных новых теорий, подходов, а также детальное рассмотрение, изложение с различных позиций существующих, с целью более глубокого понимания происходящих процессов, предвидения будущих изменений. Необходимым условием развития систем искусственного интеллекта автор полагает максимальную степень детализации, изучения используемых представлений, преобразование в явную форму очевидной, подразумеваемой, используемой по умолчанию информации.

Библиография

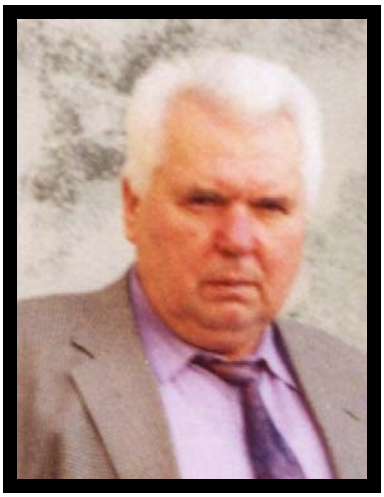
[Фано, 1965] Р. Фано. Передача информации. Статистическая теория связи. – М.: Мир, 1965. – 547 с.

[Луц, 2006] В.К. Луц. Современные тенденции развития информационных технологий. // Штучний інтелект, 2006, №.4

Информация об авторе:

Луц Василий Константинович *– Институт кибернетики им. В.М. Глушкова НАН Украины, 03680 г. Киев, пр. Глушкова, 40, Украина; e-mail: lv1@ukr.net*

In memoriam



д.т.н., профессор Геннадий Михайлович Бакан

(1936-2005)

25 июля 2005 года на 70-ом году ушел из жизни известный специалист в области информатики, технической кибернетики, теории систем управления доктор технических наук, профессор Бакан Геннадий Михайлович.

Баканом Г.М. и его сотрудниками в течение более двух десятков лет, начиная с работы в Институте кибернетики Национальной академии наук Украины, были созданы робастные методы и алгоритмы эллипсоидального оценивания состояния непрерывных и дискретных систем в условиях неопределенности свойств этих систем и внешней среды. Алгоритмы нашли применение при обработке информации, полученной в космических экспериментах, а также в системах навигации и управления космическими аппаратами. Для динамических процессов в дискретном времени с математическими моделями в классе точечно-множественных отображений решена задача аналитического синтеза точечных и множественных оценок сглаживания, фильтрации и прогноза.

С 1996 по 2000 год Бакан Г.М. работал на должности заместителя директора по научной работе Института космических исследований Национальной академии наук Украины и Национального космического агентства Украины. Под его руководством разработаны принципиальные основы обработки научной космической информации и организации вычислительного процесса в создаваемом Центре обработки научной информации.

Светлая память о Бакане Геннадии Михайловиче, как о честном, порядочном человеке, о талантливом ученом и педагоге, авторе многих научных и научно-педагогических трудов, профессоре факультета кибернетики Киевского национального университета имени Тараса Шевченко, научном руководителе аспирантов и докторантов, активном участнике конференций KDS, навсегда останется в сердцах тех, кто его знал и работал рядом с ним.

10th Anniversary of

Association for the Development of the Information Society

Acad. G. Bonchev St., block 8, Sofia 1113, Bulgaria
Tel. (+359-2) 979-3813, -3808, Fax (+359-2) 739-808
e-mail: ario@math.bas.bg, adis@einet.bg
http://www.adis.org

The Association for the Development of the Information Society (ADIS) was established in April 1997 and is an independent, non-government, non-profit organization with the non-commercial objective to support the development of the information society in Bulgaria. This objective is extensively defined in the Association's statute and includes:

- Interaction with individuals and organizations working for the development of the information society in Bulgaria and in the world.
- Support of the comprehensive utilization of the capacity of the information infrastructure and information technologies by all layers of society and all ages and professions, as well as by unemployed, ethnic minorities, people with disabilities, etc.
- Development and implementation of national and international projects whose goal is establishing, developing, and governing the information society.
- Participation in the elaboration and implementation of educational, promotional, and demonstration programs dedicated to information society issues.
- Participation in international activities on issues of the development of the information society, and maintenance of ties to and interaction with foreign and international organizations.
- Organization of conferences, forums, workshops dedicated to the information society.
- Publishing of a newsletter distributed among the individual and collective members of the Association.

Besides individual persons, the Association has as collective members from various regions of Bulgaria: Plovdiv University 'Paisii Hilendarski'. Technical University—Gabrovo, the Police Academy, the Institute of Mathematics and Informatics, the Institute of Information Technologies, the Central Laboratory of Computer Security of the Bulgarian Academy of Sciences (Sofia), and other organizations. Societies in the cities of Plovdiv, Shoumen, and Bourgas have been formed as autonomous subsidiaries of the Association. Its membership and associated structures are growing quickly and already include foreign members. The Association has existed since recently but it unites people and organizations with several decades of experience in the field of computer science and information technologies. Since 1999, the Association has organized monthly national seminars in the framework of the Forum Global Information Society. The seminars are devoted to the development of the information society in all fields of the human activities and aspects. Other activities include implementing a project for training disabled (deaf) people to use computers and the Internet, a project for training secondary school teachers in a broad range of computer technologies, participation in the drafting of the Bulgarian national strategy for the Information Society, drafting of models and principals for creating, management and development of public centers for access to Internet, information and communication services and public e-information and e-services for the Bulgarian citizens as well as delivering of talks on Information Society issues at various national and regional events by members of the Association.

The Association gladly welcomes contacts with organizations from abroad whose activities are related to the development of the global information society.

15th Anniversary of

ADUIS

ASSOCIATION OF DEVELOPERS AND USERS OF INTELLIGENT SYSTEMS

ADUIS consists of about one hundred members including ten collective members. The Association was founded in Ukraine in 1992. The main aim of ***ADUIS*** is to contribute to the development and application of the artificial intelligence methods and techniques. The efforts of scientists engaged in ***ADUIS*** are concentrated on the following problems: expert system design; knowledge engineering; knowledge discovery; planning and decision making systems; cognitive models designing; human-computer interaction; natural language processing; methodological and philosophical foundations of AI.

Association has long-term experience in collaboration with teams, working in different fields of ***research and development***. Methods and programs created in Association were used for revealing regularities, which characterize chemical compounds and materials with desired properties. Some thousands of high precise prognoses have been done in collaboration with chemists and material scientists of Russia and USA.

Association can help ***businessmen*** to find out conditions for successful investment taking into account region or field peculiarities as well as to reveal user's requirements on technical characteristics of products being sold or manufactured.

Physicians can be equipped with systems, which help in diagnosing or choosing treatment methods, in forming multi-parametric models that characterize health state of population in different regions or social groups.

Sociologists, politicians, managers can obtain the Association's help in creating generalized multi-parametric "portraits" of social groups, regions, enterprise groups. Such "portraits" can be used for prognostication of voting results, progress trends, and different consequences of decision making as well.

Association provides a useful guide in technical diagnostics, ecology, geology, and genetics.

ADUIS has at hand a broad range of high-efficiency original methods and program tools for solving analytical problems, such as knowledge discovery, classification, diagnostics, prognostication.

ADUIS unites the creative potential of highly skilled scientists and engineers

Since 1992 ***ADUIS*** holds regular conferences and workshops with wide participation of specialists in AI and users of intelligent systems. The proceedings of the conferences and workshops are published in scientific journals. ***ADUIS*** cooperates through its foreign members with organizations that work on AI problems in Russia, Byelarus, Moldova, Georgia, Bulgaria, Czechia, Germany, Great Britain, Hungary, Poland, etc. ***ADUIS*** is the collective member of the European Coordinating Committee for Artificial Intelligence (ECCAI).

Products developed by ADUIS: **Confor**: Tools for Knowledge Discovery, Classification, Diagnostics and Prediction; **Analogy**: Tools for Solving Problems on the Basis of Analogy; **Manager**: Tools for Decision Support Systems Design; **Discret**: Tool for Discretization of Numerical Data; **Gobsec**: The System for Investment Scheduling.

For contacts: V.M.Glushkov Institute of Cybernetics; National Academy of Science of Ukraine;

Prospect Akademika Glushkova, 40, 03680 GSP Kiev-187, Ukraine;

Phone: (380+44) 5262260; Fax: (380+44) 5263348; E-mail: glad@aduis.kiev.ua

60th Anniversary of

INSTITUTE OF MATHEMATICS AND INFORMATICS of Bulgarian Academy of Sciences

Acad. G. Bonchev Str., block 8, Sofia 1113, Bulgaria
Tel. (+359-2) 979-3824, Fax (+359-2) 971-3649
http://www.math.bas.bg

The Institute of Mathematics and Informatics (IMI) at BAS was founded in 1947 as Institute of Mathematics. At the beginning about ten research fellows were working at the Institute. In 1961 a computational centre was established as part of the Institute. Later specialist in Mechanics also worked at the Institute, hence and it was named Institute of Mathematics and Mechanics. Its present name dates from 1995

The Institute has considerable achievements in the field of Mathematics that are not discussed here.

The development of the Informatics in Bulgaria started at the Institute. Many researchers have built the career of Informatics specialists.

The Institute was the first in Bulgaria to buy an universal analog computing machine MH-7. The first Bulgarian computer was created at the Institute. Soon after that came into exploitation the first imported into Bulgaria computer "MINSK-2". An original software for this computer – auto code "MIKOD", operation systems "MID" and "MID-2", a system for symbol programming "MIKS" and a rich library of programs were created here as well.

The fellows of the Institute also carried out the first Informatics researches in Bulgaria. The Institute has a wide range of activities in Applied Informatics and it continues to produce original software for the solving important problems. Researchers from the Institute organized and taught the first courses in Informatics at the Sofia University "St. Kliment Ohridski" for students in Mathematics. In a short time a major in Informatics was launched with the help of the Institute and later on it became a specialty at the Sofia University. Researchers of the Institute prepared the first syllabus, textbooks, and manuals. The staff of the Institute is also involved in training teachers in Informatics for the secondary school.

In the course of the years the informaticians at focused upon the research activities and many of them are still lecturing Informatics at a number of Bulgarian universities.

Departments of IMI : *Algebra; Artificial Intelligence; Biomathematics; Complex Analysis; Differential Equations; Education in Mathematics and Informatics; Geometry and Topology; Information Research; Laboratory of Mathematical Linguistic; Logic; Mathematical Foundations of Informatics; Mathematical Linguistics; Mathematical Physics; Computational Mathematics; Operation Research; Probability and Statistics; Real and Functional Analysis; Software Engineering; Telecommunications Department.*