ITHEA

International Journal

INFORMATION TECHNOLOGIES & KNOWLEDGE

2008 Volume 2 Number 1

International Journal
INFORMATION TECHNOLOGIES & KNOWLEDGE
Volume 2 / 2008, Number 1

IJ ITK is official publisher of the scientific papers of the members of
the ITHEA International Scientific Society,
the Association of Developers and Users of Intellectualized Systems (ADUIS)
and the Association for Development of the Information Society (ADIS)

IJ ITK rules for preparing the manuscripts are compulsory.
The rules for the papers for IJ ITK as well as the subscription fees are given on *www.foibg.com*.
The camera-ready copy of the paper should be received by e-mail: info@foibg.com.
Responsibility for papers published in IJ ITK belongs to authors.

General Sponsor of IJ ITK is the Consortium FOI Bulgaria (www.foibg.com).

International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE" Vol.2, Number 1, 2008

Edited by the Institute of Information Theories and Applications FOI ITHEA, Bulgaria,
in collaboration with the V.M.Glushkov Institute of Cybernetics of NAS, Ukraine, and
the Institute of Mathematics and Informatics and the Institute of Information Technologies, BAS, Bulgaria.

Publisher: Institute of Information Theories and Applications FOI ITHEA
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, www.foibg.com, e-mail: info@foibg.com
Printed in Bulgaria

ISSN 1313-0455 (printed) ISSN 1313-048X (online) ISSN 1313-0501 (CD/DVD)

PREFACE

Intelligo ut credam !

The International Journal "Information Technologies and Knowledge" (IJ ITK) has been established in 2007 as independent scientific printed and electronic media. IJ ITK is edited by the Institute of Information Theories and Applications FOI ITHEA in collaboration with the leading researchers from the: *Institute of Cybernetics "V.M.Glushkov", NASU* (Ukraine)*; Institute of Mathematics and Informatics, BAS* (Bulgaria)*; Institute of Information Technologies, BAS* (Bulgaria)*; University of Hasselt* (Belgium)*; Natural Computing Group (NCG) of the Technical University of Madrid* (Spain)*; Astrakhan State Technical University* (Russia)*; Taras Shevchenko National University of Kiev* (Ukraine)*; University of Calgary* (Canada)*; VLSI Systems Centre, Ben-Gurion University* (Israel).

The main scope of the IJ ITK covers but is not limited to the theoretical research, applications and education in the area of the Information Technologies for:

- Knowledge Collecting and Accumulation;
- Knowledge Discovery and Acquisition;
- Knowledge Level Modeling;
- Knowledge Management, Transfer and Distributing;
- Knowledge Market;
- Knowledge Representation and Processing;
- Knowledge Utilization;
- Knowledge-based Society;
- Knowledge-based Systems.

Many scientific and practical areas are connected to the topics of interest of IJ ITK: Business Informatics: e-Management, e-Finance, e-Commerce, e-Banking, Business Intelligence: Methodology, Tools and Technologies, Analytics and Statistics; Cognitive science; Competitive Intelligence; Data Mining; Decision Making; e-Management in Governmental and Municipal Structures: Models, Systems, e-Government, etc.; Information Technologies in Biomedicine; Intelligent Communication Technologies and Mobile Systems; Intelligent Robots; Intelligent Systems; Intelligent Technologies in Control and Design; Modern (e-) Learning Information Technologies; Multimedia Semantic Systems; P2P e-Learning Applications; Planning and Scheduling; Socio-cognitive engineering; Technology and Human Resource Issues; Technology-based Blended, Distance and Open Learning; Web-based Technologies and Systems, AI/Semantic Web.

More information about the IJ ITK rules for preparing and submitting the papers as well as how to take out a subscription to the Journal may be obtained from www.foibg.com.

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

Only for two years, IJ ITK became as well-known international journal. Till now, including this volume, more than *150* papers have been published. IJ ITK authors are widespread in *20* countries all over the world: *Azerbaijan, Belarus*, *Belgium, Bulgaria*, *Canada*, *Egypt*, Finland, France, *Germany*, India, *Iran*, *Ireland*, *Israel, Japan*, Jordan, *Romania, Russia*, *Spain*, *Ukraine*, and USA

Volume 2/2008 of the IJ ITK contains 82 papers written by 194 authors from 15 countries (marked in italics above), selected from several international conferences, seminars and workshops organized or supported by the Journal. At the first place, the main source for selection were the ITA 2007 Joint International Events on Informatics, (June 17- July 8, 2007, Varna, Bulgaria)

- XIII-th International Conference "Knowledge-Dialogue-Solution" (KDS 2007);
- V-th International Conference "Information Research and Applications" (i.TECH 2007);
- Second International Conference on Modern (e-) Learning (MeL 2007);
- International Conference on e-Management & Business Intelligence (eM&BI 2007);
- VI-th International Workshop on General Information Theory (GIT 2007);
- Second Int. Workshop on Cyber Security (CS 2007).

Several papers were selected from the pool of papers directly submitted to IJ ITK.

The success of IJ ITK belongs to the whole of the ITHEA International Scientific Society. Due to great interest to IJ ITK this Volume 2 contains one complementary issue - Number 5.

We express our thanks to all authors, editors and collaborators who had developed and supported the International Journal on Information Technologies and Knowledge.

Congratulations to Prof. Volodimir Donchenko (Ukraine) who was awarded by the International Prize "ITHEA" for the year 2007. The "ITHEA" Prize has been established in 1995. It is aimed to mark the achievements in the field of the information theories and applications.

Krassimir Markov
IJ ITK Founder and Editor in chief

International Prize "ITHEA"

Awarded Scientists till 2007:

1995	*Sandansky*	*K. Bankov, P. Barnev, G. Gargov, V. Gladun, R. Kirkova, S. Lazarov, S. Pironkov, V. Tomov*
1996	*Sofia*	*T. Hinova, K. Ivanova, I. Mitov, D. Shishkov, N. Vashchenko*
1997	*Yalta*	*Z. Rabinovich, V. Sgurev, A. Timofeev, A. Voloshin*
1998	*Sofia*	*V. Jotsov*
1999	*Sofia*	*L. Zainutdinova*
2000	*Varna*	*I. Arefiev, A. Palagin*
2001	*St.Peterburg*	*N. Ivanova, V. Koval*
2002	*Primorsko*	*A. Milani, M. Mintchev*
2003	*Varna*	*T. Gavrilova, A. Eskenazi, V. Lozovskiy, P. Stanchev*
2004	*Varna*	*B. Kokinov, T. Vamos*
2005	*Varna*	*L.F. de Mingo, M. Dobreva*
2006	*Varna*	*J. Castellanos, G. Totkov*
2007	*Kiev*	*V. Donchenko*

INFORMATIONAL MODEL OF NATURAL LANGUAGE PROCESSING

Aleksandr Palagin, Viktor Gladun, Nikolay Petrenko, Vitalii Velychko, Aleksey Sevruk, Andrey Mikhailyuk

Abstract: The formal model of natural language processing in knowledge-based information systems is considered. The components realizing functions of offered formal model are described.

Keywords: natural language processing.

ACM Classification Keywords: I.2.7 Natural Language Processing - Text analysis

Architecture of modern knowledge-based information systems (KIS) with natural language knowledge representation and processing explicitly includes ontological constituent, that in general can be interpreted as a conceptual knowledgebase. Such a knowledgebase is represented as a directed graph, which vertexes are represented as frames describing concepts, and arcs are the set of conceptual relations connecting among themselves concepts. Other important feature of the specified architecture is division and separate processing of the first and second order semantics [1], that generally means separating internal language and non language processing [2] and transition to the formal-logic representation of the source text.

The specified architecture features of modern KIS transform traditional natural language texts (NLT) processing model to the following formal model

$F = \langle T, W, SS^1, O, S^2, I \rangle$, where

T is a set of processed NLT;

W is a set of word forms contained in the T;

SS^1 is a set of first-order syntactic-semantic structures, describing T;

O is a set of ontological structures, converting the sets W and SS^1 into the S^2;

S^2 is a set of second-order semantic structures, describing the set of the T's scripts;

I is a set of information-code representations of the S^2.

Let us describe the objects of the formal model.

The T set represents the corpus of the NLT described with business, scientific and technical styles.

The $W \to SS^1$ chain in its classical sense represents grammatical analysis of NLT. In contrast to traditional linear and strong coded analyzing methods we use mixed one. The gist of its that in lexicographical database full W set is represented in the tables of two types: the tables of lexemes with corresponding morphological, syntactic, and semantic characteristics and the tables of inflections for all full-meaning varying parts of speech.

At the same time the algorithms of lexemes paradigm generation are simple; the lexical tables contains lexemes' stems and corresponding codes for selecting records from the tables of inflections. Non inflection changes are considered by corresponding algorithms.

The described grammatical analysis structure univocally corresponds to an effective mapping of functional operators into hardware realization, which is in particular based on PLIS (programmable logical integral schemas).

The set O of ontological structures in an ideal represents language-ontological pattern of the world, described in [1, 3].

The SS^1 set is iteratively formed and interpreted by the syntactic-semantic subsystem like "Konspekt" [4]. The main operation of syntactic-semantic analysis is recognizing of syntactical and semantic relations, linking text words. The recognition of links between notional words is carried out by the analysis of inflections and prepositions basing on lexical models without explicit traditional grammar rules using. For each sentence in source text parsing tree is built. The solving of semantic ambiguity is carried out in the way of calling to the set of ontological structures O. Basing on built semantic trees categorial net is built. The net represents semantic space S^2 of the text.

As a computer representation of such space it is convenient to use the growing semantic net for set of information code representations I, that is organized as pyramidal net, which receptors are corresponding to the names of objects, classes of objects, properties, states, actions, relations, semantic cases, modifiers [5].

The information conversion chain $T \to W \to SS^1$ and $O \to S^2 \to I$, per se, represents (accordingly) base procedures of analysis and understanding NLT, which interpretation tools are grammatical and semantic processors.

In applications for searching and processing large volume of text documents expedient is to use knowledge-based search system [6] providing initial and final stages of documents processing – searching in the Internet and saving documents in a database in the form of their synopses generated by "Konspekt" subsystem.

The described model of natural language texts processing in knowledge-based information system, containing "Konspekt" subsystem as a part, is a promising line of development ontological-based informational systems that make active use of ontology of natural language lexicon.

Bibliography

1. Palagin O.V., Petrenko M.G. Model' kategorial'nogo rivnya movno-ontologichnoi kartyny svitu // Matematychni mashyny I systemy – 2006. – N 3. – pp.91-104.
2. Rubashkin V.Sh. Predstavlenie I analiz smysla v intellektual'nyh informatsionnyh sistemah. – M.: Nauka, 1989. – 191p.
3. Palagin A.V. Organizatsiya I funktsii "yazykovoy" kartiny mira v smyslovoy interpretataii EYa-soobshcheniy // Information Theories and Application. – 2000. – Vol. 7, N 4. pp.155-163.
4. Gladun V.P., Velichko V.Yu. Konspektirovanie estestvenno-yazykovyh tekstov. Proceedings of the XI-th International Conference "Knowledge-Dialogue-Solution"(KDS'2005).- Varna, Bulgaria.-2005.- 5. pp.344-347 vol.2.
5. Gladun V.P. Planirovanie resheniy. – Kiev: Naukova dumka, 1987. – 168p.
6. Sevruk O.O., Petrenko M.G. Znannya-orientovana poshukova systema na osnovi movno-ontologichnoi kartyny svitu // Tezy dopovidey XIII mizhnarodnoi konferencii "Avtomatyka-2006". – Vinnytsya. – 2006. – 25-28 veresnya. – p.413.

Authors' Information

Aleksandr Palagin – V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: palagin_a@ukr.net

Victor Gladun – V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: glad@aduis.kiev.ua

Nikolay Petrenko, Aleksey Sevruk, Andrey Mikhailyuk – V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: petrng@ukr.net

Vitalii Velychko – V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: vitaly@aduis.kiev.ua

HARDWARE-BASED AND SOFTWARE-BASED SECURITY IN DIGITAL RIGHTS MANAGEMENT SOLUTIONS

Maria Nickolova, Eugene Nickolov

Abstract: The main requirements to DRM platforms implementing effective user experience and strong security measures to prevent unauthorized use of content are discussed. Comparison of hardware-based and software-based platforms is made showing the general inherent advantages of hardware DRM solutions. Analysis and evaluation of the main flaws of hardware platforms are conducted, pointing out the possibilities to overcome them. The overview of the existing concepts for practical realization of hardware DRM protection reveals their advantages and disadvantages and the increasing demand for creation of multi-core architecture, which could assure an effective DRM protection without decreasing the user's freedom and importing risks for end system security.

Keywords: Security, DRM protection.

ACM Classification Keywords: D.4.6 Security and Protection.

Introduction

Security design is one of the most challenging areas for system designers because it requires an extraordinary effort to build a system offering strong security features but not hindering the working process of users and being well accepted by them. This is particularly true as far as the compromise between the content owner's copyrights and the right of free access and exchange of information is concerned. The solution adopted in last decade is the digital rights management. Although most users don't agree with the use of DRM, it is of critical importance for authors, publishers and content providers - their business depends on the ability to control and to charge for access to their content.

Although the inherent insecurity of Internet, many upper-layer security protocols can be used to protect data during transmission but content is still at risk when it arrives at its destination. If the end device's boot process and critical information are not highly secure, the digital content can be stolen after the transmission and distributed without permission. This implies that end user devices must be built on a trusted platform and equipped with mechanisms for cryptographically validating the hardware environment and code signatures of downloaded software [1].

The DRM technologies allowing the protection of the content by access from unauthorized users could be divided into three groups: DRM implemented completely by software, DRM implemented completely by hardware, and the hybrid combinations of software and hardware. Certainly the most secure DRM is that which is implemented by hardware, the next most secure is the hybrid, and the least secure is via software.

Main requirements to DRM platforms

An effective DRM technology must provide a smooth and effective user experience for content use and in the same time must implement strong security measures to prevent unauthorized use of content [2]. The main requirements to it are:

1. It must ensure fully protected capabilities, which means the protection functions should be performed as part of the boot process. Otherwise during boot-up malicious software can easily hook the control functions and compromise system integrity. If end devices receive content over a network, such malicious software could be masked as a firmware upgrade or Trojan, or hidden using rootkits.

2. It must allow trusted integrity measurement and confirmation, that means the platform should own the capability to automatically check in real time during the boot all the new software and executable files in the system (certificates, digital signatures). Once this confirmation is done, the operating system loader can be started and the boot process proceeds as normal.

3. It must provide integrity reporting to notify the user about the results of the integrity measurement and possibly to prevent the user from playing back the DRM protected content in case of negative results from the integrity check.

Obviously these requirements could be implemented by hardware and/or software means.

Advantages of hardware-based DRM versus software-based

The analysis of the commercially available technologies for DRM protection shows two main reasons to use hardware-based security of the protected content: better overall robustness and improved user experience. The main benefits of the hardware-based security robustness are:

- Immunity from the inherent vulnerabilities and security holes of the used operating system. The security of all software applications is limited by the level of security provided by the underlying OS. Although the open and rich OS have bigger security challenges than a closed OS a hardware security module is an essential element to make the OS trustworthy.
- Impossibility to access, change or uninstall security features. Attacks to DRM protection often start by targeting the protection software - trying to uninstall it or stop its activity [3]. Obviously hardware-based DRM protection cannot be uninstalled as it is hard coded into the chips.
- Protected memory. Hardware-based DRM solutions manage the memory in a restricted manner and are able to prohibit access to it, providing better protection against attacks on the security mechanism. Software solutions use memory by the services of the operating system and several processes can access the same memory space simultaneously. Most OS provide some memory protection, but the safety of the memory space depends on the extent to which the operating system is robust and free of flaws. This is particularly important for the cryptographic algorithms which require the storage of the intermediate results during the execution of the cryptographic module. If the content of this temporary storage is exposed, the entire DRM system can be easily compromised.
- Better performance. The hardware DRM protection could be optimized for maximum security and operate independently, not degrading the performance of the computer or consuming its resources.
- Prevention of potential software conflicts. The software DRM protection is run on the same computer with many other security programs using together the same processor, memory, OS and other resources. This could provoke various conflicts resulting in poor performance and even in stopping the action of both DRM protection software and security programs.
- Secure Storage. Hardware-based DRM protection is able to better protect sensitive data, such as private keys. A software DRM implementation cannot prevent the exposure of keys and therefore they could be relatively easily compromised. Even very strong cryptographic algorithms could be easily compromised by an direct or indirect attack to their software implementation. Only a proper hardware implementation, to which countermeasures against known attacks are added, could protect the secrecy and the integrity of the DRM mechanism.
- True Random Number Generation. The software DRM technologies use pseudo-random numbers that decrease the security level of the DRM protection. As random numbers are used in DRM protection process for the creation of temporary and special values and are part of challenge response authentication, the better the random number generator, the more secure the DRM implementation.
- Easier, faster and cheaper attacks to software DRM solutions. This is related to the security vulnerabilities, which are inherent for software modules and to the presence of many hackers who have enough time, knowledge and wish for breaking the relevant protection.
- Quick dissemination. The compromising of software DRM solutions by only one hacker becomes quickly available for general use. The publishing in Internet of correspondent methodology allows it to be used by a lot of end-users before the manufacturer could take measures to remove the vulnerabilities in the protection, and to bring severe damage to operators, content providers and manufacturers.
- Less susceptibility to reverse engineering. Hardware-based platforms are able to apply special measures that hide the data-dependent fluctuations in power consumption while software-based DRM solutions are more vulnerable to attacks based on power analysis.

- Most content applications like music, video and games require efficient and effective user experience which is the key factor for the success of consumer electronic devices and therefore for the acceptance of DRM by users. The main benefits of the hardware-based improvement of user experience are:
- Superior performance in which user experience is prioritized without sacrificing security. Hardware solutions generally accelerate several times cryptographic functions (which are computation-intensive) in comparison with software solutions, making DRM security operations almost invisible for the end-user.
- Optimization of CPU power and memory use. Although the computing power of modern processors increases constantly and should allow relatively fast handling of cryptographic functions, processors are designed mainly for new demanding applications such as video rendering and high quality graphics. Therefore software-based cryptographic operations are able to overload them and to worsen the user experience. There are some cryptographic operations (exchange of protocols with long keys, for example) that affect inadmissibly user experience.
- Improved power consumption and memory use. The use of hardware-based DRM platforms allows the CPU to operate at a lower clock rate, saving power which is particularly important for battery powered mobile devices. Additionally, software-based solutions require more memory (code size needs large buffers) which affect the speed and the quality of other applications.

Disadvantages of hardware-based DRM platforms

- Software modification or creation of new software on a computer with hardware-based DRM technology may require hardware changes that could be slow and expensive.
- The simple replacing of a peripheral device running protected content could cause a hardware-based DRM system to refuse to run software.
- Network cards replacement could make a computer unusable until other necessary hardware modifications are done and passwords are reauthorized. This process may require the cooperation of several vendors.
- If DRM protection is compromised reinstalling is impossible.
- Manufacturers of hardware-based DRM are not able to warrant that DRM agents or their hardware assistants will not cause or help any safety or security failures.
- More difficult implementation of extended usability. Software DRM implementations facilitate the making of the licensed content usable by a user anywhere in his personal network (local hard drive, media center, iPOD, cell phone, home entertainment center or burning to a CD), for the hardware-based it's more difficult and expensive.
- Higher security in hardware-based DRM solutions means higher costs, less interoperability, longer development cycles and potentially shorter market life.
- Limited flexibility. It's difficult to make hardware DEM systems open to new uses, new business models, or new rights created by content owners.

It is clear that these flaws could be easily overcame and that only a hardware-based DRM implementation or a hybrid hardware/software solution could address all required security challenges while allowing seamless user experience [4].

Approaches for implementing hardware-based DRM

Two main concepts have been developed by now: trusted system concept and multi-core concept.

Trusted system concept

The Trusted Computing Group (TCG), successor of the Trusted Computing Platform Alliance (TCPA), is an initiative led by AMD, IBM, Intel, Hewlett-Packard, Microsoft, Sony, and Sun Microsystems. Its aim is to develop and promote open, vendor-neutral, industry standard specifications for trusted computing building blocks and software interfaces across multiple platforms [5]. The new principles in the TCG architecture expand the range of entities that are able to use TCG features as a trust basis. These entities could include not only the direct user of the platform and the owner but also some remote entities wishing to interact with this platform. The TCG architecture introduces the mechanism of remote attestation which allows remote third parties to ask a platform

for details of its current software state. On the basis of the attestation made, third parties can decide to consider the platform's configuration as trustworthy or not. If correctly implemented, this kind of remote attestation could become an important feature for DRM clients on open platforms as it may help a content provider when he makes a decision about the reliability of the client before the content is actually provided. What makes TC technology especially attractive for implementing DRM is its ability to enforce usage policies. Once their security conditions are violated, TC systems stop working. Since their security conditions are built as a "chain of trust" [6] containing hardware-locked keys and certificates from trusted third parties, they are hard to modify, at least much harder than software-based systems. If a DRM solution relies on a trusted system, it is easy to implement a hard-to-break usage rights management chosen by content owners. TC technology is not necessary or sufficient to implement DRM but it can make implementing DRM easier and cheaper. An example of such a realization is the Intel Wireless Trusted Platform with the Certicom Security Architecture software. In this technology a special trusted platform module is built directly into the processor and provides secure key and password storage and protection. First, a secure boot process authenticates the hardware platform and the security architecture authentication module, then the module runs DRM applications and allows the users to access DRM protected content. The security architecture requires decoding the keys using information stored in secure hardware, to be able to access the content, after what these keys are used to decrypt and use the content, but only on the specific device. The encrypted content is not locked to this device, because another user is not able to use the content without paying to the content provider for having access to the rules for the content use [7].

Multi-core concept

Intel's Hyper-Threading technology allows parallel processing at thread-level on a single-core processor by sharing the processor's resources. In Intel multi-core processor, each thread is processed independently by a separate dedicated processor, which allows full parallel execution at hardware-level and software-level and is very suitable for DRM applications [8].

In 2005 Intel embedded DRM capabilities within its dual-core processor Pentium D and allowed (theoretically) copyright holders to prevent unauthorized use and distribution of DRM protected materials [9]. But some functional problems with the distribution of jobs in the cores and in the chip-set when both cores are enabled caused applications to crash or hang and finally made hardware DRM capabilities unusable for real protection of content. The next stage in the implementation of hardware-based DRM in Intel's products was Lenovo's ThinkPad model, launched in 2006. It uses a combination of fingerprint sensor, trusted platform module chip and special software from Microsoft and Adobe to control access and distribution only of PDF documents.

AMD also planned to incorporate DRM into future GPUs by blocking the access to the frame buffer and allowing access only to certain software from certain vendors but these plans didn't involve AMD multi-core processors because of the complexity of problems in sharing and synchronizing DRM-related actions.

In 2006 IBM announced the technology Secure Blue intended for use in digital media players, electronic organizers, mobile phones, computers and devices where data is encrypted and decrypted as it runs through the processor and maintained encrypted in the device's RAM. Secure Blue requires a few circuits to be added to any processor design in order to enforce strict access controls at the hardware level.

Conclusion

It is obvious that DRM is becoming an integrated part of any copyright protected intellectual product in digital form and therefore DRM protection should be implemented in hardware and/or software assuring highest stability and performance as well as the best copyright protection possible. Different adopted solutions have many advantages and disadvantages but clearly show that it is impossible to realize well working solutions based only on software tools. Hardware-based platforms, especially those using multi-core processors demonstrate really promising features by improving user experience along with the robustness of DRM protection. Technology from a hardware standpoint is already in place, thanks to the efforts of various chipset manufacturers who have driven an evolution to support the benefits of parallel processing. Now research must be conducted to develop suitable multi-CPU architectures and multithreaded software that will guarantee the building of the perfect DRM system – fast, flawless and cheap - that can be neither broken nor avoided.

Bibliography

[1] Biddle, P., England, P., Peinado, M. and Willman, B. (2003). The Darknet and the future of content protection. In Digital Rights Management-Technological, Economic, Legal and Political Aspects.LNCS 2770, Springer.

[2] CEN/ISSS, (2003). Digital Rights Management Report http://europa.eu.int/comm/enterprise/ict/policy/doc/drm.pdf.

]3] M. Nickolova, E. Nickolov, Verification and Application of Conceptual Model and Security Requirements on Practical DRM Systems in E-Learning. In: First International Workshop "Cyber Security" - CS 2006.

[4] Hauser, T. and Wenz, C. (2003): DRM Under Attack: Weaknesses in Existing Systems. In Digital Rights Management-Technological, Economic, Legal and Political Aspects. LNCS 2770, Springer.

[5] Peinado, M. Chen, Y. et al. (2004), NGSCB: A Trusted Open System. In Proceedings of 9th Australasian Conference on Information Security and Privacy ACISP, Sydney, Australia, July 13-15.

[6] Smith, S.W. (2005): Trusted computing platforms: Design and applications. Berlin, Heidelberg, New York: Springer.

[7] Dornan, A. (2006): Yes, trusted computing is used for DRM; Information Week, 17 February 2006.

[8] Rump, N. (2003): Digital rights management: Technological aspects. In: Becker et al. (2003).

[9] Pakman, D. (2005): Why DRM everything? A sensible approach to satisfying customers and selling more music in the digital age; Groklaw, 31 December 2005.

Authors' Information

Maria Nickolova – National Laboratory of Computer Virology, BAS, Acad.G.Bonthev St., bl.8, Sofia-1113, Bulgaria; e-mail: maria@nlcv.bas.bg.

Eugene Nickolov - – National Laboratory of Computer Virology, BAS, Acad.G.Bonthev St., bl.8, Sofia-1113, Bulgaria; e-mail: eugene@nlcv.bas.bg.

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).

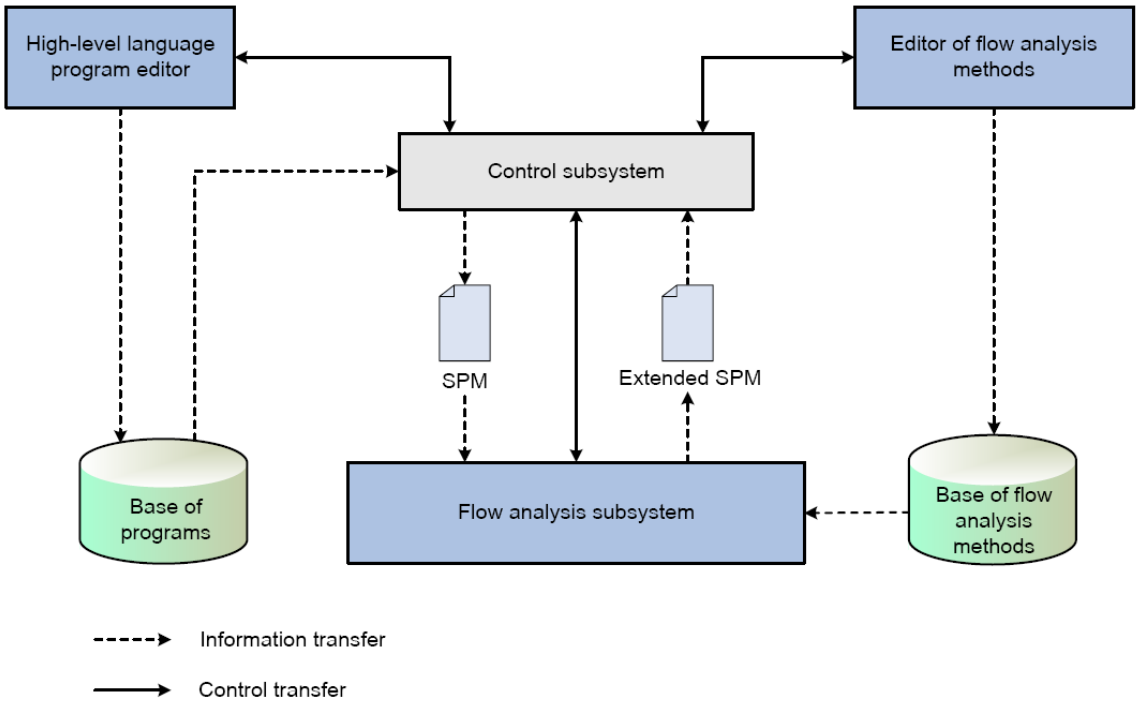


Fig. 1. Architecture of subsystem of knowledge-managed flow analysis

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@imcs.dvgu.ru, vd2000@mail.ru

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>.”

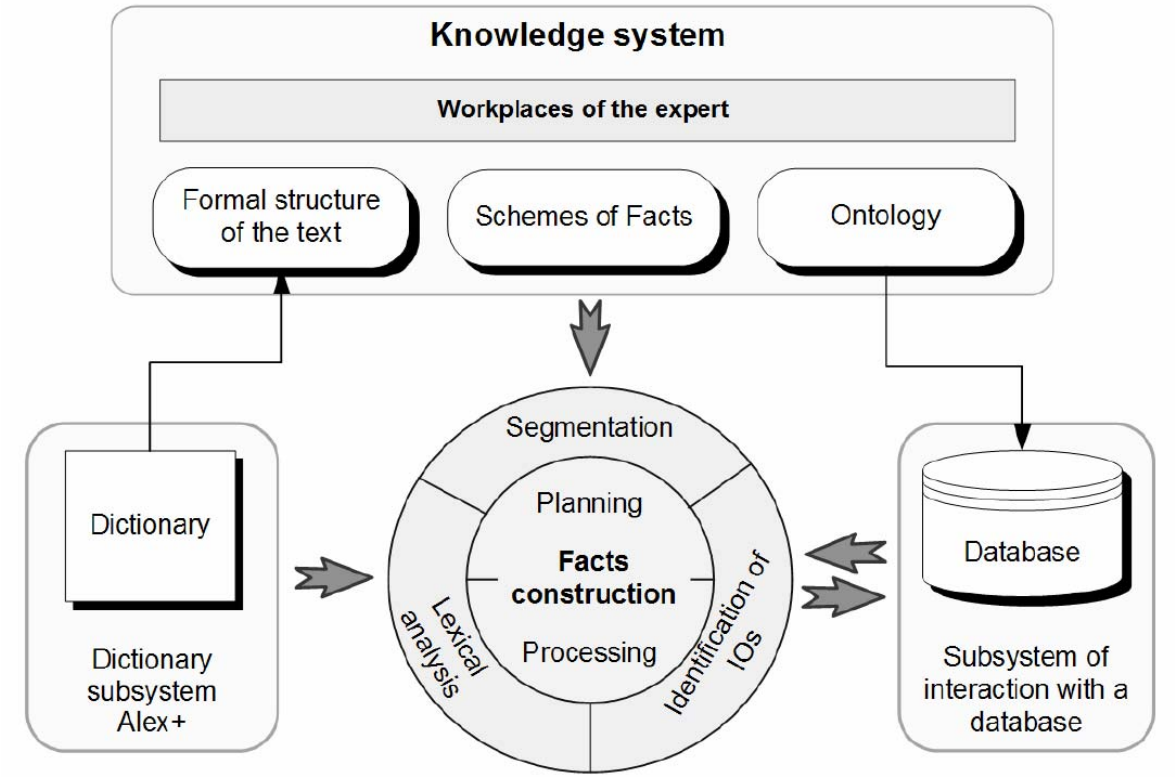
The following facts have been extracted from this sentence:

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

AN EFFECTIVE METHOD FOR CONSTRUCTING DATA STRUCTURES SOLVING AN ARRAY MAINTENANCE PROBLEM

Adriana Toni, Angel Herranz, Juan Castellanos

Abstract: In this paper a constructive method of data structures solving an array maintenance problem is offered. These data structures are defined in terms of a family of digraphs which have previously been defined, representing solutions for this problem. We present as well a prototype of the method in Haskell

Keywords: array maintenance, average complexity, data structures, models of computation

Introduction

The Range Query Problem of size N (N-RQP) deals with the analysis and design of data structures for the implementation of the operations Update and Retrieve: let A be an array of length N of elements of a commutative semigroup, Update(i, x) increments A(i) (i-th element of A) in x and Retrieve(i, j) outputs the partial sum A(i)+..+A(j).

In [4] we find the following definition of N-RQP design.

Definition N-RQP design

A N-RQP design is a triple (Z, U, R) where Z is an array of length M with N less or equal M, U is a family of subsets of 1..M indexed on 1..N and R is a family of subsets of 1..M indexed on 1..N × 1..N. Given a N-RQP design (Z, U, R), the implementation of the operations Update and Retrieve is:

```
procedure Update
    (i: 1..N, x:S) is
begin
    for k in U_i loop
      Z(k) := Z(k) + x;
    end loop;
end Update;
```

```
function Retrieve
    (i: 1..N, j: 1..N)
    return S is
begin
    return  ∑_{k∈R_ij} Z(i)
end Retrieve;
```

It is a well known result that an N-RQP design (Z, U, R) is a N-RQP solution if and only if

$$\forall i, j, k \in 1..N \bullet \left|R_{ij} \cap U_k\right| = \begin{cases} 1 \text{ if } i \le k \wedge k \le j \\ 0 \text{ otherwise} \end{cases}$$

and a proof can be found in [1].

In [4] we find the three definitions below as well.

Definition N-RQP graph

An acyclic digraph G =(V, E) is a N-RQP graph if the following conditions hold:

(1) V=1..M with N≤M.

(2) For every vertex v≤N, its out-degree is 0.

(3) For every vertex v>N, Successors(G,v)∩1..N≠0.

Definition N-RQP design in terms of G

Given a N-RQP graph G =(V, E), the N-RQP design (Z, U, R) is a N-RQP design in terms of G if it verifies the following properties:

(1) $\left|Z\right| = \left|V\right|$

(2) $U_i = Ancestors^{\bullet}(G,i)$

(3) R_{ij} *is the set of vertices with the smallest cardinal that verifies:*

$$\bigcup_{u \in R_{ij}} Successors^{\bullet}(G,u) \cap 1..N = i..j$$

$$\bigcup_{u \in R_{ij}} Successors^{\bullet}(G,u) \cap 1..N = 0$$

being

$$Successors^{\bullet}(G,u) = \{u\} \cup Successors(G,u)$$

$$Ancestors^{\bullet}(G,v) = \{v\} \cup Ancestors(G,v)$$

and in the same paper it has been proved that given a N-RQP graph, a N-RQP design in terms of G is a N-RQP solution.

Definition 2^K -RQP graph

Let K be a natural number. A 2^K -RQP graph G^K is defined inductively:

(1) $If \quad K = 0 \quad then \quad G^K = (\{(1,1)\}, 0)$

(2) $If \quad K > 0 \quad then \quad G^k = Duplicate(G^{K-1})$

where function Duplicate is defined as

```
function Duplicate (G^K = (V^K,E^K) : Digraph) return Digraph is
  N : constant ℕ := 2^K
  M : constant ℕ := |V^K|
  V : {(i, j) ∈ 1..N × 1..N • i ≤ j} := 0;
  E :  P(V×V) := 0;
  i, j : 1..(2N);
begin
  -- The ''cloning'' loops
  for (i, j) in V^K loop
    V := V ∪ {(i, j), (i + N, j + N)};
  end loop;
  for (i, j) → (i', j') in E^K loop
    E := E ∪ {(i, j) → (i', j'), (i + N, j + N) → (i' + N, j' + N)};
  end loop;
  -- (V,E) is a graph with two subgraphs which are just like G
  -- but with different node numbering
  for i in 1..(N−1) loop -- The ''left half'' loop
    j := i + 1;
    while (i,N) ∉ V ∧ j ≤ N loop
      if (i, j) ∈ V ∧ (j, N) ∈ V then
        V := V ∪ {(i, N)};
        E := E ∪ {(i, N) → (i, j), (i,N) → (j,N)};
      else
        j := j + 1;
      end if;
    end loop;
  end loop;
  for j in (N + 2)..(2N) loop -- The ''left half'' loop
    i := j − 1;
    while (N + 1, j) ∉ V ∧ i ≤ 2N loop
      if (N, i) ∈ V ∧ (i, j) ∈ V then
        V := V ∪ {(N, j)};
        E := E ∪ {(N, j) → (N, i), (N, j) → (i, j)};
      else
        i := i + 1;
      end if;
    end loop;
  end loop;
  return (V,E);
end Duplicate;
```

Obviously, the 2^K -RQP design (Z,U,R) can be computed after the construction of the 2^K -RQP graph as described by the following brute force algorithm:

Algorithm 1 *The following algorithm computes R_{ij} for a N-RQP design in terms of a N-RQP graph G=(V,E):*

```
R : P(1..|V|) := 1..N;
R' : P(1..|V|);
begin
  for R' in P(1..|V|) loop
    if |R'|≤|R|
      ∧ ⋃_{u∈R_ij} Successors•(G,u) ∩ 1..N = i..j
      ∧ ⋃_{u∈R_ij} Successors•(G,u) ∩ 1..N = 0  then
      R := R'
    end if;
  end loop;
  return R;
end;
```

The algorithm is correct for any N-RQP graph but in the case of 2^K -RQP graphs a refinement can be applied by filtering those R' with a cardinal greater than 2 reducing the complexity drasticly. Nevertheless, the user is just interested in the design and not in the graph so a direct constructive method that computes |Z|, U and R would be welcome. In this section a method for calculating 2^K -RQP designs is given..

As in the previous section, Z can be treated as a two dimensional array (where the variable Z(i, j) does not necessarily exist for all (i, j)) that is isomorphic to a one dimensional array Z' and where the isomorphism is given by an injective partial map such that $(i,j) \to i$ *when* $i=j$.

The method presented in the following definition is the result of a deep analysis of the properties of 2^K -RQP graphs.

Definition 1

Let be the 2N-RQP with $N = 2^K$ and $K \in \mathbf{N}$ A 2^{K+1}-RQP design (Z, U, R) is constructed in the following way:

$R_{i\ j}$ $(i \in 1..2N, j \in i..2N)$ ***is defined by the following cases:***

- ***A1.*** *If* $i = j$,

$$R_{i\ j} = \{(i, j)\} \tag{4}$$

- ***A2.*** *For every* $l \in 1..K$,
  - *If* $i \in 1..2^{l-1}$,

$$R_{i\ 2^l} = \{(i, 2^l)\} \tag{5}$$

and for every $c \in 1..(2^{K-l} - 1)$ *and* $d = c2^{l+1}$,

$$R_{i+d\ 2^l+d} = \{(i + d, 2^l + d)\} \tag{6}$$

  - *For every* $r \in 1..(l - 2)$,
    If $i \in (2^l - 2^{l-r} + 2)..(2^l - 2^{l-r-1})$,

$$R_{i\ 2^l} = \{(i, 2^l)\} \tag{7}$$

and for every $c \in 1..(2^{K-l}-1)$ *and* $d = c2^{l+1}$,

$$R_{i+d\ 2^l+d} = \{(i+d, 2^l+d)\} \tag{8}$$

- ***A3.*** *For every* $l \in 1..K$,
  - *If* $j \in (2^{l-1}3+1)..2^{l+1}$,

$$R_{2^l+1\ j} = \{(2^l+1, j)\} \tag{9}$$

and for every $c \in 1..(2^{K-l}-1)$ *and* $d = c2^{l+1}$,

$$R_{2^l+1+d\ j+d} = \{(2^l+1+d, j+d)\} \tag{10}$$

  - *For every* $r \in 1..(l-2)$,
    If $j \in (2^l+1+\frac{2^l}{2^{r+1}})..(2^l-1+\frac{2^l}{2^r})$,

$$R_{2^l+1\ j} = \{(2^l+1, j)\} \tag{11}$$

and for every $c \in 1..(2^{K-l}-1)$ *and* $d = c2^{l+1}$,

$$R_{2^l+1+d\ j+d} = \{(2^l+1+d, j+d)\} \tag{12}$$

- ***B1.*** *If* $i \in 1..N$ *and* $j \in N+1..2N$,

$$R_{i\ j} = \{(i, N), (N+1, j)\} \tag{13}$$

- ***B2.*** *For every* $l \in 1..(K-1)$,
  - *If* $i \in 1..2^l$ *and* $j \in (2^l+1)..(2^{l+1}-1)$,

$$R_{i\ j} = \{(i, 2^l), (2^l+1, j)\} \tag{14}$$

$$R_{2N-j+1\ 2N-i+1} = \{(2N-j+1, 2N-2^l), (2N-2^l+1, 2N-i+1)\} \tag{15}$$

  - *If* $i \in 2..2^l$ *and* $j \in (2^l+1)..(2^{l+1}-2)$, *and for every* $c \in 1..(2^{K-l}-1)$ *and* $d = c2^{l+1}$,

$$R_{i+d\ j+d} = \{(i+d, 2^l+d), (2^l+d+1, j+d)\} \tag{16}$$

U_k $(k \in 1..2N)$ ***is defined by the following comprehension set:***

$$U_k = \{(i,j) \bullet i \in 1..2N \ \wedge \ j \in i..2N \ \wedge \ i \leq k \ \wedge \ k \leq j \ \wedge \ |R_{ij}| = 1\}$$

$|Z|$***, the number of variables*** $Z(i,j)$ ***of the design is the number of*** R_{ij} ***of size*** 1***:***

$$|Z| = \sum_{|R_{ij}|=1} 1$$

Implementation in Haskell

The following Haskell [7] program implements the constructive method given in Definition 1.
This prototype implementation has been tested for $N=2^k$ being K less or equal 25.
Given an integer K, most functions compute information of the solutions of the 2^{K+1}-RQP: |Z|, U_i and R_{ij}.

```
pow2 :: Integer -> Integer
pow2 0 = 1
pow2 n = 2 * (pow2 (n-1))

a1 :: Integer -> [[(Integer,Integer)]]
a1 k = [[(i,i)] | i <- [1..(pow2 (k+1))]]
```

```
a2 :: Integer -> [[(Integer,Integer)]]
a2 k = [[(i, pow2 l)]
         | l <- [1 .. k],
           i <- [1 .. pow2 (l-1)]]
       ++
       [[(i+d, pow2 l + d)]
         | l <- [1 .. k],
           i <- [1 .. pow2 (l-1)],
           c <- [1 .. pow2 (k-l) - 1],
           let d = c * pow2 (l+1)]
       ++
       [[(i, pow2 l)]
         | l <- [1 .. k],
           r <- [1 .. l-2],
           i <- [pow2 l - pow2 (l-r) + 2 .. pow2 l - pow2 (l-r-1)]]
       ++
        [[(i+d, pow2 l + d)]
          | l <- [1 .. k],
          r <- [1 .. l-2],
          i <- [pow2 l - pow2 (l-r) + 2 .. pow2 l - pow2 (l-r-1)],
          c <- [1 .. pow2 (k-l) - 1],
          let d = c * pow2 (l+1)]
a3 :: Integer -> [[(Integer,Integer)]]
a3 k = [[(pow2 l + 1, j)]
         | l <- [1 .. k],
           j <- [3 * pow2 (l-1) + 1 .. pow2 (l+1)]]
       ++
       [[(pow2 l + 1 + d, j + d)]
         | l <- [1 .. k],
           j <- [3 * pow2 (l-1) + 1 .. pow2 (l+1)],
           c <- [1 .. pow2 (k-l) - 1],
           let d = c * pow2 (l+1)]
       ++
       [[(pow2 l + 1, j)]
         | l <- [1 .. k],
           r <- [1 .. l-2],
           j <- [pow2 l + 1 + pow2 l `div` (pow2 (r+1))
                 ..pow2 l - 1 + pow2 l `div` pow2 r]]
       ++
       [[(pow2 l + 1 + d, j +d)]
         | l <- [1 .. k],
           r <- [1 .. l-2],
           j <- [pow2 l + 1 + pow2 l `div` (pow2 (r+1))
                 ..pow2 l - 1 + pow2 l `div` pow2 r],
           c <- [1 .. pow2 (k-l) - 1],
           let d = c * pow2 (l+1)]
r1 :: Integer -> [[(Integer,Integer)]]
r1 k = a1 k ++ a2 k ++ a3 k
b1 :: Integer -> [[(Integer,Integer)]]
b1 k = [[(i,pow2 k),(pow2 k + 1,j)]
         | i <- [1 .. pow2 k],
           j <- [pow2 k + 1 .. pow2 (k+1)]]
b2 :: Integer -> [[(Integer,Integer)]]
b2 k = [[(i,pow2 l),(pow2 l + 1,j)]
         | l <- [1..k-1],
           i <- [1..pow2 l],
           j <- [pow2 l + 1..pow2 (l+1) - 1]]
       ++
       [[(pow2 (k+1) - j + 1, pow2 (k+1) - pow2 l),
         (pow2 (k+1) - pow2 l + 1,pow2 (k+1) - i + 1)]
         | l <- [1..k-1],
           i <- [1..pow2 l],
           j <- [pow2 l + 1..pow2 (l+1) - 1]]
       ++
       [[(i+d,pow2 l + d),(pow2 l + d + 1,j+d)]
```

```
        | l <- [1..k-1],
          i <- [2..pow2 l],
          j <- [pow2 l + 1..pow2 (l+1) - 1],
          c <- [1..pow2 (k-l) - 2],
          let d = c * pow2 (l+1)]

r2 :: Integer -> [[(Integer,Integer)]]
r2 k = b1 k ++ b2 k

r :: Integer -> [[(Integer,Integer)]]
r k = r1 k ++ r2 k

u :: Integer -> [(Integer,Integer)]
u k = [(i,j) | i <- [1 .. pow2 (k+1)],
               j <- [i .. pow2 (k+1)],
               i <= k, k <= j,
               (i,j) `elem` concat (a1 k ++ a2 k ++ a3 k)]

zCard :: Integer -> Integer
zCard k = fromIntegral (length (r1 k))
```

We can prove that given a N-RQP solution (Z,U,R) obtained by applying the method in Definition 1, we have:

1. The number of program variables required is

$$|Z| = N\log_2 N - 2N + 2\log_2 N + 2$$

2. The sum of costs of all update operations is

$$\frac{N^2}{2} - \frac{N}{2}\log_2 N + \frac{3N}{2} - 2$$

3. The sum of costs of all retrieve operations is

$$N^2 + N(3 - \log_2 N) - 2\log_2 N - 2$$

4. The average complexity of the Update and Retrieve operations is constant (this is a consequence of 2 and 3 above)

Bibliography

[1] D.J. Volper, M.L. Fredman, *Query Time Versus Redundancy Trade-offs for Range Queries*, Journal of Computer and System Sciences 23, (1981) pp.355--365.

[2] W.A. Burkhard, M.L. Fredman, D.J.Kleitman, *Inherent complexity trade-offs for range query problems*, Theoretical Computer science, North Holland Publishing Company 16, (1981) pp.279--290.

[3] M.L. Fredman, *The Complexity of Maintaining an Array and Computing its Partial Sums*, J.ACM, Vol.29, No.1 (1982) pp.250--260.

[4] A. Herranz, A. Toni, *Digraphs Definition for an Array Maintenance Problem*, Preprint.

[5] A. Toni, *Lower Bounds on Zero-one Matrices*, Linear Algebra and its Applications, 376 (2004) 275--282.

[6] A. Toni, Complejidad y Estructuras de Datos para el *problema de los rangos variables*, Doctoral Thesis, Facultad de Informática, Universidad Politécnica de Madrid, 2003.

[7] S. P. Jones, J. Hughes, *Report on the Programming Language Haskell 98. A Non-strict Purely Functional Language,* (February 1999).

Authors' Information

Adriana Toni – Grupo de Validacion y Aplicaciones Industriales, Facultad de Informática, Universidad Politécnica de Madrid; 28660-Boadilla del Monte, Madrid, SPAIN; e-mail: atoni@fi.upm.es

Ángel Herranz Nieva – Assistant Professor; Departamento de Lenguajes y Sistemas Informáticos; Facultad de Informática; Universidad Politécnica de Madrid; e-mail: aherranz@fi.upm.es

Juan Castellanos – Departamento de Inteligencia Artificial, Facultad de Informática – Universidad Politécnica de Madrid (Campus de Montegancedo) – 28660 Boadilla de Monte – Madrid – Spain; e-mail: jcastellanos@fi.upm.es

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 $D = \prod_{j=1}^{n} D_j$.

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).

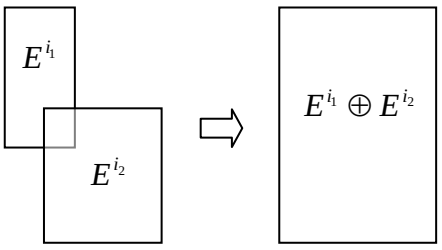


Fig. 1.

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

* The work was supported by the RFBR under Grant N07-01-00331a.

$\rho(E^1,E^2)=\sum_{j=1}^{n}k_j\rho_j(E_j^1,E_j^2)$ 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 $r^{i_1i_2}:=d(E^{i_1i_2},E^{i_1}\cup E^{i_2})$.

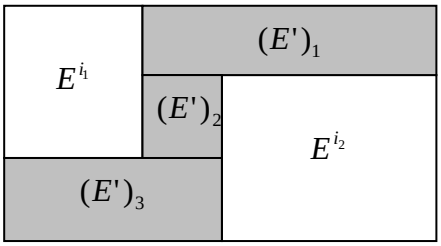


Fig. 2.

The value $d(E,F)$ is defined as follows: $d(E,F)=\max\limits_{E'\subseteq E\setminus F}\min\limits_{j}\dfrac{k_j\mid E'_j\mid}{diam(E)}$, where E' is any subset such that its projection on subspace of quantitative features is a convex set (see Fig. 2), $diam(E)=\max\limits_{x,y\in E}\rho(x,y)$.

By definition, put $I_1=\{\{1\},...,\{m_l\}\},...,\ I_q=\{\{i_1,...,i_q\}\mid r^{i_ui_v}\le\delta$ 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 $r^{i_ui_v}\le\delta$ 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.

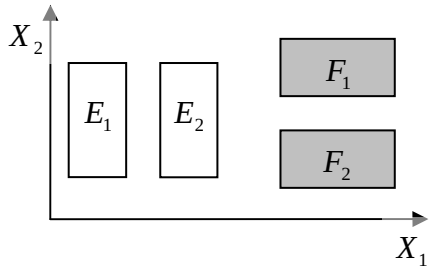


Fig. 3.

Note that we can consider another criterion of unification (instead of $r^{i_ui_v}\le\varepsilon$): 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*

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

A CIRCUIT IMPLEMENTING MASSIVE PARALLELISM IN TRANSITION P SYSTEMS

Santiago Alonso, Luis Fernández, Fernando Arroyo, Javier Gil

Abstract: Transition P-systems are based on biological membranes and try to emulate cell behavior and its evolution due to the presence of chemical elements. These systems perform computation through transition between two consecutive configurations, which consist in a m-tuple of multisets present at any moment in the existing m regions of the system. Transition between two configurations is performed by using evolution rules also present in each region.

Among main Transition P-systems characteristics are massive parallelism and non determinism. This work is part of a very large project and tries to determine the design of a hardware circuit that can improve remarkably the process involved in the evolution of a membrane. Process in biological cells has two different levels of parallelism: the first one, obviously, is the evolution of each cell inside the whole set, and the second one is the application of the rules inside one membrane. This paper presents an evolution of the work done previously and includes an improvement that uses massive parallelism to do transition between two states. To achieve this, the initial set of rules is transformed into a new set that consists in all their possible combinations, and each of them is treated like a new rule (participant antecedents are added to generate a new multiset), converting an unique rule application in a way of parallelism in the means that several rules are applied at the same time. In this paper, we present a circuit that is able to process this kind of rules and to decode the result, taking advantage of all the potential that hardware has to implement P Systems versus previously proposed sequential solutions.

Keywords: Transition P System, membrane computing, circuit design.

ACM Classification Keywords: D.1.m Miscellaneous – Natural Computing

Introduction

Transition P-systems or Membrane Computing (designed by [Păun, 1998]) are based on the processes that occur among living cells. The idea behind it is the fact that a living cell may change its state depending on the set of elements that are present in it and, of course, depending on the chemical rules that can transform them. Based on this, we can create a computational model based on that behavior. So, there is a definition of a cellular structure that contains elements that can be repeated, conforming multisets, and rules that define how multisets are combined to reach cell evolution. One of these structures (membranes) may contain another ones, conforming a hierarchical relation whose components may communicate among them, always based on what the rules allow. Evolution due to a rule application may cause that a membrane passes information to the one immediately superior in the hierarchy or to any of the ones that are in a level immediately inferior. All this, besides the fact that eventually, a membrane may be inhibited or dissolved by means of some rule application, and that they may have different priorities, does P-systems very interesting in order to define their hardware implementation.

All these processes can be viewed as computational ones and P systems have been sufficiently characterized from a theoretical point of view and their computational power has been settled. However, nowadays, the way in which these models have to be implemented is still a problem not solved. This problem is having two different approaches: software and hardware models. There are many papers about software tools implementing different P system variants [Gutierrez-Naranjo, 2006], but in the case of P-systems hardware implementation, only a few references can be found: connectivity arrays for membrane processors [Arroyo, 2004], multisets and evolution rules representation in membrane processors [Arroyo, 2004b] or a hardware membrane system description using VHDL [Petreska, 2003]. However, in [Martinez, 2006a] and [Martinez, 2006b] there is a hardware approach that implements a circuit that covers the whole process that takes place inside a membrane. Authors describe the way a sequential circuit may control the application of active rules in a Transition P –system and its internal structure.

Being aware that P-systems are defined as "distributed, massively parallel and non deterministic", we think these characteristics should be strengthen. Parallelism takes place in this model in two different levels: the first one is

due to the fact that every cell or membrane evolutions at the same time than the others, and the second one is due to the fact that rules inside each membrane may be applied at the same time.

It is at this point where this work pretends to be positioned: parallelism by means of application of multiple rule at the same time.

The structure of this paper presents, first, the problem and its methodological solution and afterwards, shows its data model and a general representation of the circuit, as well as each part in detail.

The algorithm

As we may read in [Martinez, 2006a] and [Martinez, 2006b], a hardware approach to P-system is possible. These papers show how the general algorithm of an evolution system may be developed with a circuit. Authors clearly improved the basic algorithm by the way of the proposal of obtaining the number that represents the maximum times each rule could be applied to the current multiset. This number, called *applicability MAX*, is the higher limit for a random number that indicates how many times the rule will be applied, modifying the basic algorithm as:

Let R be the initial set of active rules, R = {R_1, R_2, …, R_n } and W the initial multiset, being *input(R_i)* the antecedents for rule R_i

1. R ←InitialActiveRules
2. REPEAT
3. R_i ← Aleatory (R)
4. *MAX* ← Applicability (R_i, W)
5. IF *MAX* = 0
6. THEN R ← R - {R_i}
7. ELSE
8. K ← Aleatory(1, *MAX*)
9. W ← W – K * input(R_i)
10. count(K, R_i)
11. UNTIL |R| = 0

As we can see, the algorithm works by selecting randomly one rule until there are no rules to apply (|R| = 0). Once the rule is selected, it calculates its *MAX* value; if this value is zero, it means that the rule is no more applicable and it has to be removed from the set of rules.

Afterwards, it generates a random number *K*, equal or less than *MAX* and the application of the rule consists in subtracting *K* times the antecedents input(R_i) from W. This means that such rule is being *K* times used. Of course we have to store this value so we can check how many times a rule has been applied (step 10).

So, this algorithm is implementing some way of parallelism (in each iteration, a rule is applied K times). However, the importance of parallelism in this kind of model, as well as its possible importance in the field of NP problem solving, urged us to find a way to be able to apply several rules at the same time, improving its throughput (after all, the exposed algorithm just calculates *MAX* for one rule). Thus, the idea is to find a way to select several rules and apply them over the multiset in each evolution step. We could see that this could be achieved in a better way, improving its computational throughput just by considering the initial set of rules as a new set composed by the rules that result form calculating the power set P(R) from the original set of rules. So if we have:

$R = \{R_1, R_2, \ldots, R_n\}$

its power set is:

$P(R) = \{\varnothing, R_1, R_2, \ldots, R_n, R_1R_2, \ldots, R_1R_n, \ldots, R_{n-1}R_n, \ldots, R_1R_2 \ldots R_{n-1}R_n\}$

As {∅} is an element with no rules and it has no meaning for this work, the power set minus the empty set will be considered:

$P'(R) = P(R) - \{\varnothing\} = \{ R_1, R_2, \ldots, R_n, R_1R_2, \ldots, R_1R_n, \ldots, R_{n-1}R_n, \ldots, R_1R_2 \ldots R_{n-1}R_n\}$

If we consider now this set P'(R) as the initial active rules set, what we are doing is to be able to apply several rules at the same time, by the meaning that if a rule $R' \in P'(R)$ / $R' = R_xR_yR_z$ is chosen, a possible evolution may process the antecedents of several rules ($R_xR_yR_z$) at the same time (as many as conform the chosen element). The algorithm, right now would be:

Let R be the initial set of active rules, R = {R_1, R_2, …, R_n } and W the initial multiset, being *input(R_i)* the antecedents for rule R_i

Let P(R) be the power set of R and P'(R) = P(R) - {∅} with card(P(R)) = 2^n and card(P'(R) = 2^n -1))

```
1.  REPEAT
2.      ∀ Ri ∈ P'(R), ||        MAXi ← Applicability (Ri, W)
3.      ∀ Ri ∈ P'(R), ||        Ki ← Aleatory(1, MAXi)
4.      COBEGIN
5.          ∀ Ri ∈ P'(R), ||    WT ← Ki * input(Ri)
6.          END ← ¬ ∃ Ki <>0; IF NOT END
7.                              THEN BEGIN
8.                                  Rj ← Aleatory (P'(R)) / Ki <>0
9.                                  COBEGIN
10.                                   W ← W – WT
11.                                   count (Ki, Rj, R)
12.                                 COEND
13.                             END
14.     COEND
15.  UNTIL END
```

As we may see, this algorithm underlines the importance of parallelism, taking advantage from the processes that can be done simultaneously. As we will see ahead, there are two types of parallelism: first, some processes are applied to all the rules at the same time (indicated by the sign "||" in steps 2, 3 and 5) and second, some control processes may be done simultaneously (indicated by the clauses "COBEGIN … COEND").

Moreover, differences with the previous algorithm include (steps 2 and 3) calculating applicability *MAX* and a random number (K_i, between 1 and its *MAX* value) for each of the rules that are included in P'(R). As they should be calculated simultaneously, process time is not incremented. Once this is done, it calculates the product of each K_i by the antecedents of each rule, but, at the same time this is happening, there is a special process (steps 6 through 8) that selects a random rule but just for the rules whose *MAX* value is different than zero (this means that K_i is also different than zero). This causes that any selected rule is applicable and only in the case that no rule has *MAX* value greater than zero, the *END* condition is reached.

Once the rule is selected, of course the system has to subtract the antecedents (W_T) from the set of elements (W) but, again at the same time, it has to decode the participant rules, because not all the rules that are in P'(R) appear also in R. A rule could be the result from the composition of several rules from R and so, the process has to increase the counter for each of the rules from R.

The model and data representation

Before we can start with the circuit design, there is the need for a definition of a data structure that contains information about the initial membrane state, the initial multiset of objects and the set of evolution rules. Continuing with the work done in precedent papers, and knowing that we have to establish some limits for a suitable circuit, the model should:

a. Limit the cardinality O = {a, b, c, d, e, f, g, h, i, j} of the alphabet to 10.
b. Define the initial multiset involved in a specific membrane *i*, W_i, that will be represented by a 4-bits register. The length of this register will be 10. The value in each register position will represent the number of occurrences for the object represented by the alphabet letter in that position.
c. The finite set of evolution rules R associated to the membrane *i* is represented by a set of registers, each of ones represents the antecedents of rule *i*, and the value in each position represents the element occurrences needed for the current rule to be applied.
d. The Application Rules Register is represented by a register which length is, at least, $\log_2 n$, being n = card(P(R))

In this work we have to consider two main aspects:

a. First, the initial set of rules is considered to be the power set of active rules at the beginning of the process. The circuit to obtain active rules from the initial multiset may be obtained from [Martinez, 2006a].
b. Shown solution will be scalable, so, increasing number of initial rules will not have a negative influence in the design (if card(R)=n, then card(P(R)) = 2^n and card(P'(R) = 2^n -1)). In this paper we will work with examples with a set of three initial rules, that makes card (P'(R)) = 7.

The circuit shown in figure 1 takes the set of rules, already P'(R) members (*Initial Active Rules*), and the initial multiset of objects and brings out a complete register (*Application Rules Register*) with the occurrences each rule should be applied to obtain a step of evolution.

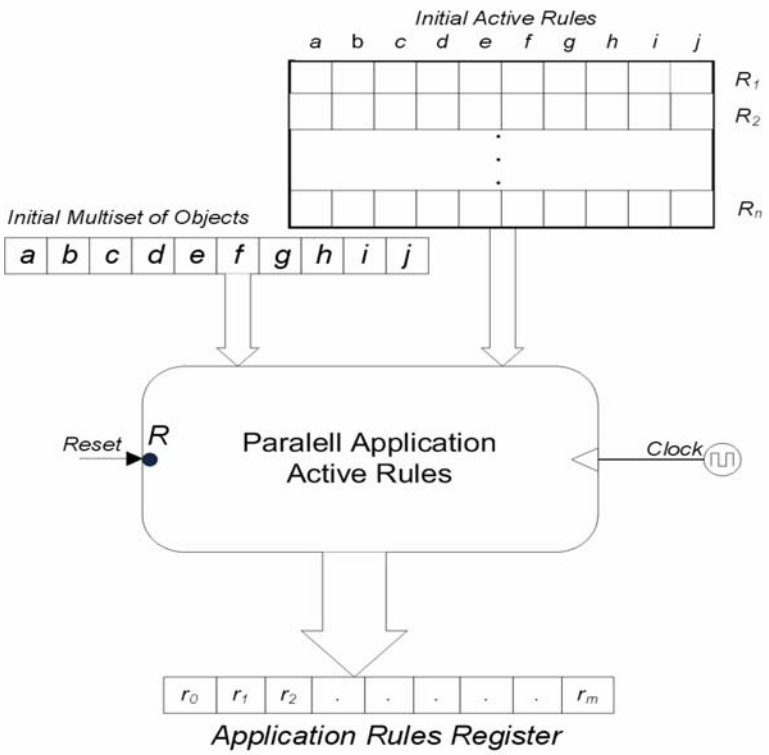


Figure 1. Circuit inputs and outputs

The circuit

The circuit is the result for assembling different functional units created each one to do a specific job. All of them should be coordinated by a "Logic Control Unit" not represented in figure 2, that takes the control and repeats the whole cycle until the signal provided by the Application Selector F.U. indicates that are no more active rules.

The different units are:

Applicability MAX F.U.: This functional unit is the one that receives an active rule and determines its *Aplicability Max* value, as explained before. This value is calculated as the largest number of times current rule can be applied without having in mind the other rules. So, this functional unit needs, as input, the antecedents of current rule and the multiset of objects. The output will be the MAX value for current rule.

The Max value may be obtained [Martinez, 2006a] by dividing each position value from the register for antecedents by its corresponding position value in the multiset register. Once obtained all this results, the smallest one will be the maximum value the rule may be applied.

Random generator 1..MAX: once the Applicability Max is obtained, the circuit should generate randomly a value for each of the available rules in the active rules register. This value represents the number that each rule should be applied in case that specific rule is chosen to be the one that consumes the elements and its lowest value will be 1 and the highest will be Max_i.

It is very important to realize that Max value could be zero, due to the fact that a rule could not be applied because there are no enough elements in the multiset. In another type of circuit, this would cause the rule to be invalid for the process and that could be a problem. In this case, this is solved by the Application Selector F.U. where a k not equal to zero is selected.

If n is the cardinality of P'(R), all this calculation (n times a random number between 1 and Max_i) may be done at the same time, forcing the higher parallelism.

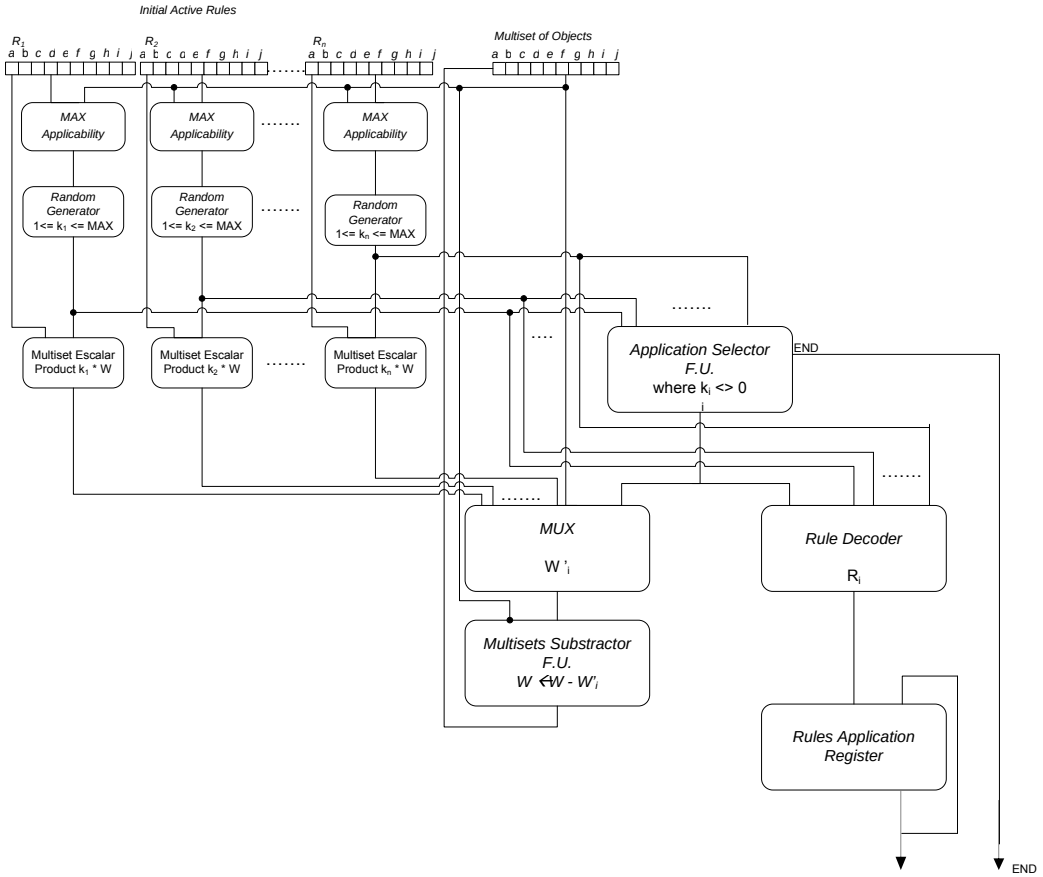


Figure 2: Circuit functional units

Application Selector F.U.: Once the previous random generator has calculated k_i for each rule, we can find that any of these numbers may be zero (due to the fact that its *Application Max* value may be equal to zero). We have to implement a way of avoiding to choose a rule with k_i equal to zero because it would cause a delay time in process dedicated, probably, to recalculate a new k_i. To avoid this kind of problems, we developed a functional unit that can generate a random number but just for those rules which k is greater than zero.

Achieving the developing of this functional unit included developing of one special cell that obtains the position of the first "1" appearing in the register, and another cell to get the position of the second "1", and another for the third, and so on. We will have as many cells as number of rules in P'(R). As result of this, we will get together all the positions that have a value for k_i different from zero.

As we can see in figure 3, to do this, first we need to transform k values, that can be greater than one, to another values (1 or 0) representing that k_i has a value greater than zero or not. This can be done with a comparator.

Thus, there is a need to have a specific circuit to detect the first “1” in the register, that would be the position of the first rule that has a non zero value. In figure 3 we can see that there is a comparator that sets the position of the value “1” by deactivating the logical gates after it finds the value. Comparison with values 1 to 7 brings us the value of the position for the first rule that has a non zero value for the random number k. There has to be another specific circuit to detect the value for the second rule, the third, etc. Of course, these circuits are similar to the one shown but they ignore the registers behind the position they are looking for.

If we call each of these circuits A, B, C, D…, all of them should be added as we can see in figure 4, in such a way that the first values are all different from zero. Now, all we have to do is to generate a random value no greater than the position of the last number greater than zero. To achieve this, we just have to add the number of values different from zero that are stored in the register and use it as the input for the random generator. The output will be a number between 1 and the number of values different from zero. If we use it as the index for the multiplexer, we will obtain always a value indicating the position of a rule which random k is different from zero and so, we are sure the rule is applicable and active.

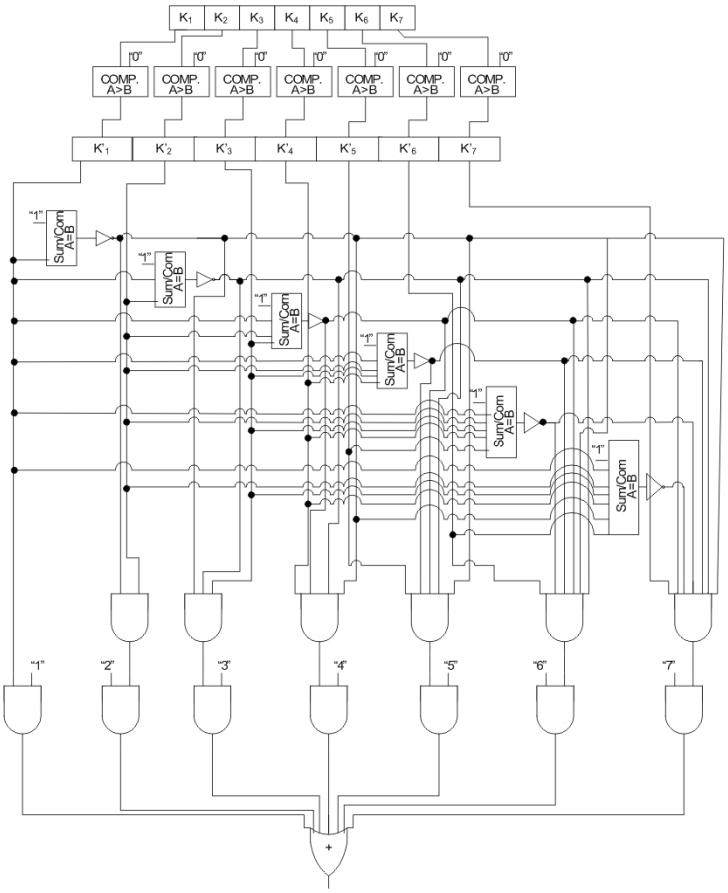


Figure 3: Detecting first rule with $k_i > 0$

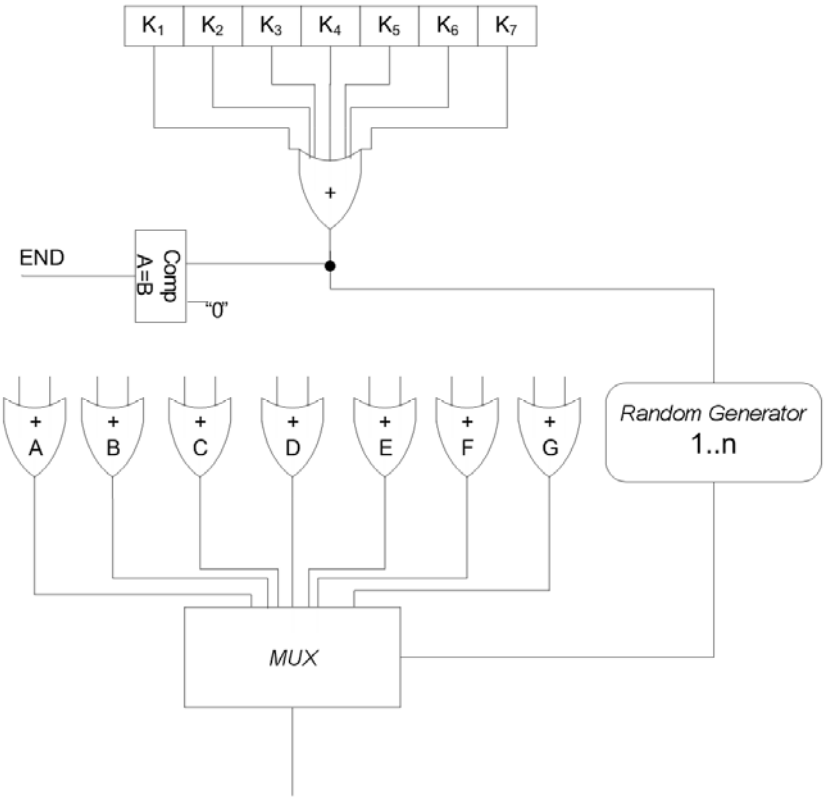


Figure 4: End signal and output for random generator $k_i <> 0$

Of course, if no k value is different from zero, the addition would result in a zero value, which, once compared with “0”, results in the "*END*" signal for all the circuit because it means that no more rules are applicable.

Rule decoder: As we can see in figure 1, once a rule is chosen by the *Application Selector F.U.*, we need to perform two different processes: the first one is to calculate the occurrences of elements used to be able to decrement them from the multiset of objects. But there is still another problem: we should be able to register in the *Application Rules Register* the number of times each rule was applied. This means that if the rule applied i was one that belonged to P’(R) but was not in R (possible due to the way we conformed P’(R)), we have to “decode” that rule to the set of rules that conformed R. Circuit in figure 5 shows how it can be done.

The first set of comparators select the rule indicated by the functional unit “*Application Selector*”. As we just have seen, the value for the selected rule, k_i, can not be zero. Once we this value, we have to separate the components that conform it.

So, in the example with 3 rules for R ({ R1, R2, R3}) and 7 rules in P’(R):

P’(R) = { R_1, R_2, R_3, R_1 R_2, R_1R_3, R_2R_3, $R_1R_2R_3$}

If rule 1 is selected, the circuit will add only the k value for this rule (gate at the left), but if rule 7 is selected, then it will add the k value for rules R_1, R_2 and R_3 because rule 7 is $R_1R_2R_3$ and all of them were applied k times.

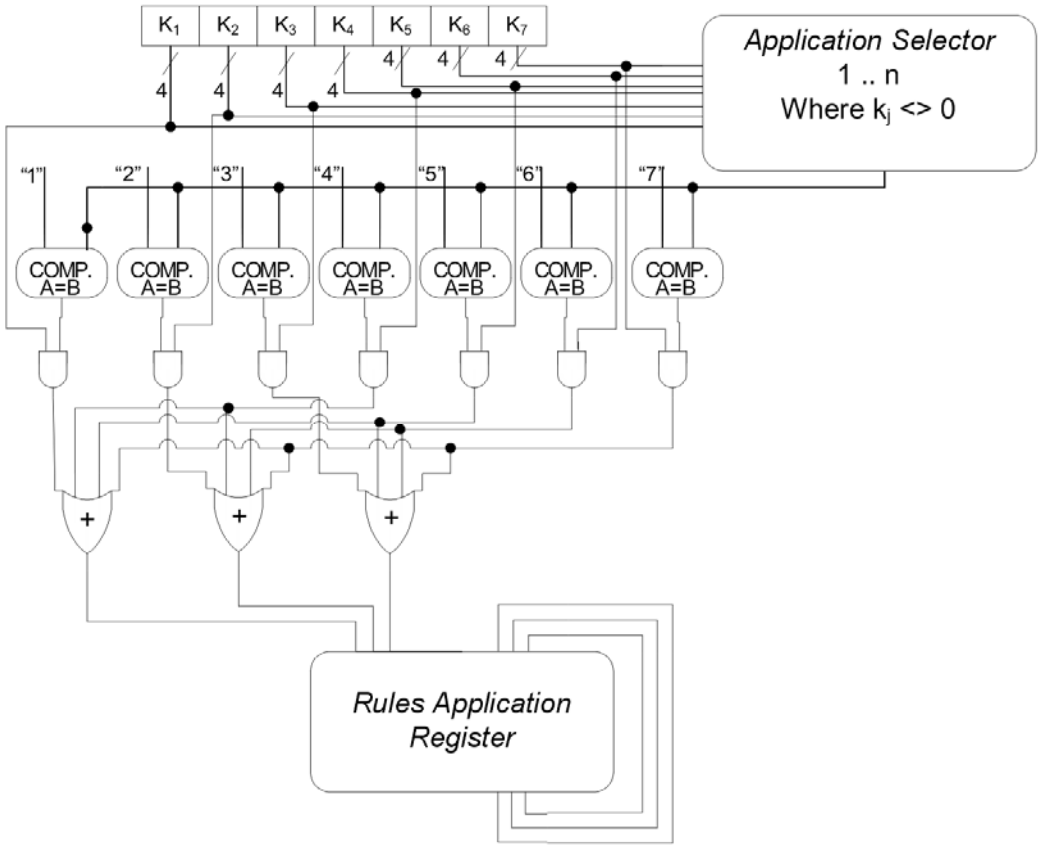


Figure 5: Rule decoder

Whenever the *Application Selector F.U.* enables de *END* signal the "Rules Application Register" will contain the final result, that is, the number of times each rule has to be applied to go forward with a transition. This number is referring the initial set of rules R.

Other functional units: Of course, there are more functional units that are in charge of calculating the final amount of elements that the circuit used during each step of evolution. There is a unit that is dedicated to calculate, for each rule, the result of multiplying k (random number generated by the first generator) by each *input(R_i)* (elements in the antecedent of each rule). Of course this can be done for all the rules at the same time and it just needs a multiplier per rule.

The second one is just a multiplexer in charge of receiving the rule number (j) selected by the *Application Selector F*.U. and to select, according with it, the product k_j * *input*(R_j).

Once this is done, we need just to decrement the product selected before from the global multiset, and this is the job for the last functional unit, storing its result in the Multiset Register of Objects to allow a new selection of a rule and let whole process go again.

Conclusion

Nowadays there are several projects trying to conform different types of circuits to implement membrane computational model with hardware, obtaining active rules and forcing the system to evolution and obtain the number of rules applied. This paper presents how to improve this kind of circuits by emphasizing the massive parallel character P-systems have.

The circuit provides the number of times each rule should be applied to do a complete transition between two configurations, according to its initial set of rules and initial multiset of objects. Of course, different applications over the same sets, do not have to produce the same result.

Hardware implementation is based on basic components like registers, counters, multiplexers, logical gates and so on. The development of the system can be done using hardware-software architectures like VHDL and physical implementation may be accomplished on hardware programmable devices like FPGA's.

Bibliography

[Arroyo, 2004a] F. Arroyo, C. Luengo, Castellanos, L.F. de Mingo. A binary data structure for membrane processors: Connectivity Arrays. A. Alhazov, C. Martin-Vide, G. Mauri, G. Paun, G.Rozenberg, A. Saloma (eds.): Lecture Notes in Computer Science, 2933, Springer Verlag, 2004, 19-30.

[Arroyo, 2004b] F. Arroyo, C. Luengo, Castellanos, L.F. de Mingo. Representing Multisets and Evolution Rules in Membrane Processors. Pre-proceedings of the Fifth Workshop on Membrana Computing (WMC5). Milano, Italy. June 2004, 126-137.

[Gutiérrez-Naranjo, 2006] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Nez. Available membrane computing software. In G. Ciobanu, Gh. Paun, M.J. Pérez (eds.) Applications of Membrane Computing.Berlin, Germany. Springer Verlag, 2006. pp.411-436. ISBN: 3-540-25017-4.

[Martínez, 2006a] V. Martinez, L.Fernández, F.Arroyo, I.García, A. Gutierrez. A HW circuit for the application of Active Rules in a Transition P System Region. Proceedings on Fourth International Conference Information Research and Applications (i.TECH-2006). Varna (Bulgary) June, 2006. pp. 147-154. ISBN-10: 954-16-0036-0.

[Martínez, 2006b] V. Martínez, L. Fernández, F. Arroyo, A. Gutierrez. HW Implementation of a Bounded Algorithm for Application of Rules in a Transition P-System. Proceedings on 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC-2006). Timisoara (Romania) septiembre, 2006. pp. 32-38.

[Păun, 1998] Gh.Păun. Computing with membranes. Journal of Computer and System Sciences, 61 (2000), and Turku Center for Computer Science-TUCS Report No 208, 1998.

[Păun, 1999] Gh.Păun. Computing with membranes. An introduction. Bulletin of the EATCS, 67, 139-152, 1999.

[Petreska, 2003] B. Petreska and C. Teuscher. A hardware membrane system. A. Alhazov, C. Martin-Vide, Gh. Paun (eds.): Pre-proceedings of the workshop on Membrane Computing Tarragona, July 17-22 2003, 343-355.

Authors' Information

Santiago Alonso Villaverde – Natural Computing Group of Universidad Politécnica de Madrid. - Dpto. Organización y Estructura de la Información de la Escuela Universitaria de Informática, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: salonso@eui.upm.es

Luis Fernández Muñoz – Natural Computing Group of Universidad Politécnica de Madrid. - Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: setillo@eui.upm.es

Fernando Arroyo Montoro – Natural Computing Group of Universidad Politécnica de Madrid. - Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: farroyo@eui.upm.es

Javier Gil Rubio – Natural Computing Group of Universidad Politécnica de Madrid. - Dpto. Organización y Estructura de la Información de la Escuela Universitaria de Informática, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: jgil@eui.upm.es

A HIERARCHICAL ARCHITECTURE WITH PARALLEL COMUNICATION FOR IMPLEMENTING P SYSTEMS

Ginés Bravo, Luis Fernández , Fernando Arroyo, Juan A. Frutos

Abstract: Membrane systems are computational equivalent to Turing machines. However, its distributed and massively parallel nature obtain polynomial solutions opposite to traditional non-polynomial ones.

Nowadays, developed investigation for implementing membrane systems has not yet reached the massively parallel character of this computational model. Better published approaches have achieved a distributed architecture denominated "partially parallel evolution with partially parallel communication" where several membranes are allocated at each processor, proxys are used to communicate with membranes allocated at different processors and a policy of access control to the communications is mandatory. With these approaches, it is obtained processors parallelism in the application of evolution rules and in the internal communication among membranes allocated inside each processor. Even though, external communications share a common communication line, needed for the communication among membranes arranged in different processors, are sequential.

In this work, we present a new hierarchical architecture that reaches external communication parallelism among processors and substantially increases parallelization in the application of evolution rules and internal communications. Consequently, necessary time for each evolution step is reduced. With all of that, this new distributed hierarchical architecture is near to the massively parallel character required by the model.

Keywords: Architecture, hierarchy, P systems

ACM Classification Keywords: D.1.m Miscellaneous – Natural Computing

Introduction

Possibilities offered by Natural Computation and, specifically P-Systems, for solving NP-problems, have made researchers concentrate their work towards HW and SW implementations of this new computational model. Transition P systems were introduced by [Păun, 1998]. They were inspired by "basic features of biological membranes". One membrane defines a region where there are a series of chemical components (multisets) that are able to go through chemical reactions (evolution rules) to produce other elements. Inside the region delimited by a membrane can be placed other membranes defiining a complex hierarchical structure that can be represented as a tree. Generated products by Chemical reactions can remain in the same region or can go to another region crossing a membrane. As a result of a reaction, one membrane can be dissolved (its chemical elements are transferred to the container membrane) or can be inhibited (the membrane becomes impermeable and not let objects to pass through).

Membrane systems are dynamics because chemical reactions produce elements that go through membranes to travel to other regions and produce new reactions. This dynamic behaviour is possible to be sequenced in a series of evolution steps between one system configuration to another. These system configurations are determined by the membrane structure and multisets present inside membranes. In the formal Transition P systems model can be distinguished two phases in each evolution step: rules application and communication. In application rules phase, rules of a membrane are applied in parallel to the membrane multiset inside of it. Once application rules phase is finished, then it begins communication phase, where those generated multisets travel through membranes towards their destination in case it is another region. These systems carry out computations through transitions between two consecutive configurations, what turn them into a computational model with the same capabilities as Turing machines.

Power of this model lies in the fact that the evolution process is massively parallel in application rules phases as well as in communication phase. The challenge for researchers is to achieve hardware and/or software implementations of P systems respecting the massively parallelism in both phases. The goal of this work is to design a new hierarchical communication architecture that approaches the best possible way to the inherent characteristics of P systems: application and communication massively parallel.

This paper is structured in the following way: in the first place, the related works are enumerated analyzing the proposed architectures, next a communication hierarchical architecture model is introduced stating detailed analysis of the model. Afterward a comparative analysis with other architectures is presented and finally the conclusions obtained are presented.

Related Works

In [Syropoulos, 2003] and [Ciobanu, 2004] distributed P systems implementations are presented. They use respectively, the Java Remote Method Invocation (RMI) and the Message Passing Interface (MPI) over a PC cluster's Ethernet network. These authors don't make a detailed analysis about importance of time spent in communication phase respect total time of P system evolution, although Ciobanu declares that "the response time of the program has been acceptable. There are however executions that could take a rather long time due to unexpected network congestion" [Ciobanu, 2004].

In reply to this problem, [Tejedor, 2007] presents an analysis of an architecture named "partially parallel evolution with partially parallel communication". This architecture is based on the following pillars:

a. Membranes distribution. At each processor, *K* membranes are allocated that will evolve, at worst, sequentially. Where,

$$K = \frac{M}{P} \quad , K \geq 1 \tag{1}$$

and *M* is the total number of membranes of the P system and *P* is the number of processors of the distributed architecture. The physical interconnection of processors is made through a shared communication line. In this scenario, there are two sorts of communications,

- internal communications that are the ones that occur between membranes allocated at the same processor, and whose communication times is negligible because they are carried out using shared memory techniques.
- external communications that are those that occur between different processors because the membranes that needs to communicate are in different processors.

The benefit obtained is that the number of the external communications decreases.

b. Proxy for processor. Membranes that are in different processors do not communicate directly. They do by the means of proxys hosted at their respective processor. Proxys are used to communicate among processors. A proxy assumes communications among membranes of one processor towards the proxy of another one. In the same way, when information from other proxys is receive, it is redistributed to the membranes of the processor.

The benefit of using proxys in the communication among membranes instead of direct communication occurs because the communication protocols penalize the transmission of small packets due to protocol overhead. So, communicate N messages of L length is slower than one message of (S * L) length.

c. Tree topology of processors. The benefit obtained by using a tree topology in the processors interconnection is that the total number of external communications is minimized due to proxys only communicate with their direct ancestor and direct descendants. This way, total number of external communications is 2(P-1).

d. Token passing in the communications. In order to avoid collision and network congestion, it has been established and order in the communication. The idea is not to have more than one proxy trying to transmit at the same time.

The analysis of this distributed architecture leads to the following conclusions:

- This solution avoids communication collisions and reduces the number and length of the external communications.
- In this model, minimum time for an evolution step (T_{min}) is determined by the formula:

$$T_{\min} = 2\sqrt{2\,M\,T_{apl}\,T_{com}} - 2T_{com} \tag{2}$$

where, T_{apl} is the maximum time used by the slowest membrane in applying its rules, and T_{com} is the maximum time used by the slowest membrane for communication

- The number of processors (P_{opt}) that leads to the minimum time is:

$$P_{opt} = \sqrt{\frac{T_{apl}\, M}{2 T_{com}}} \tag{3}$$

Hierarchical Architecture

Previous model parallelize over P_{opt} processors the application of rules and the internal communications among membranes in the same processor. On the other hand, external communications, necessaries for the communication among membranes allocated at the same processor, are sequential. For that reason, we propose a variation that permits to parallelize, up to a certain degree, external communications among nodes. This way, time of an evolution step is reduced drastically and it will tend towards the massively parallel character of a P system.

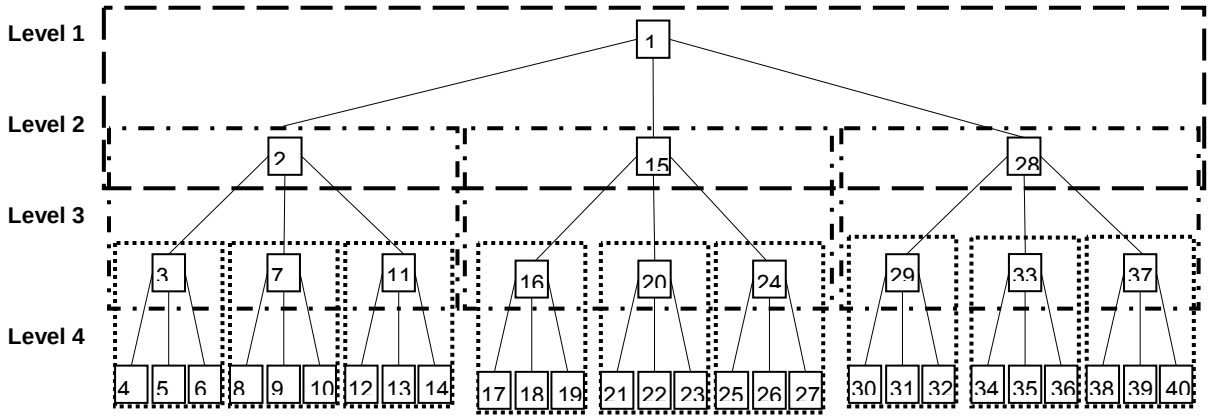


Figure 1. Hierarchical Architecture of 4 levels and amplitude equal to 3.

The new architecture consists of having at its distribute the processors in a hierarchical way, specifically, in a balanced tree of *N* levels depth and *A* processors in amplitude. For instance, figure 1 shows a balanced tree of *N* = 4 and *A* = 3.

For example of figure 1, when every node have applied its rules in parallel, external communications are carried out sequentially in each one of the 9 subtrees arranged between levels 3 and 4; hence, at every instant, as many external communications are carried out as subtrees exist between levels 3 and 4. Subsequently, external communications in each one of the three subtrees arranged between levels 2 and 3 are carried out sequentially; hence, at every instant, as many external communications are carried out as subtrees exist between levels 2 and 3. And finally, external communications in the subtree arranged between levels 1 and 2 are carried out sequentially.

From a logical point of view, each subtree requires a particular physical network to reach the parallelism of its external communications. This way, the processors of intermediate subtrees need 2 communication interfaces, one for the network of the subtree which is root, and another one for network of the subtree which is a leaf. On the other hand, only one interface is required for the processors in the extreme levels 1 and N because they are part of just one subtree. On the other hand, from a physical point of view, the number of logical networks can be reduced to one using Ethernet switches because they permit the separation of collision domains.

Figure 2 chronogram shows the parallelism in application times and in the external communications of the previous example.

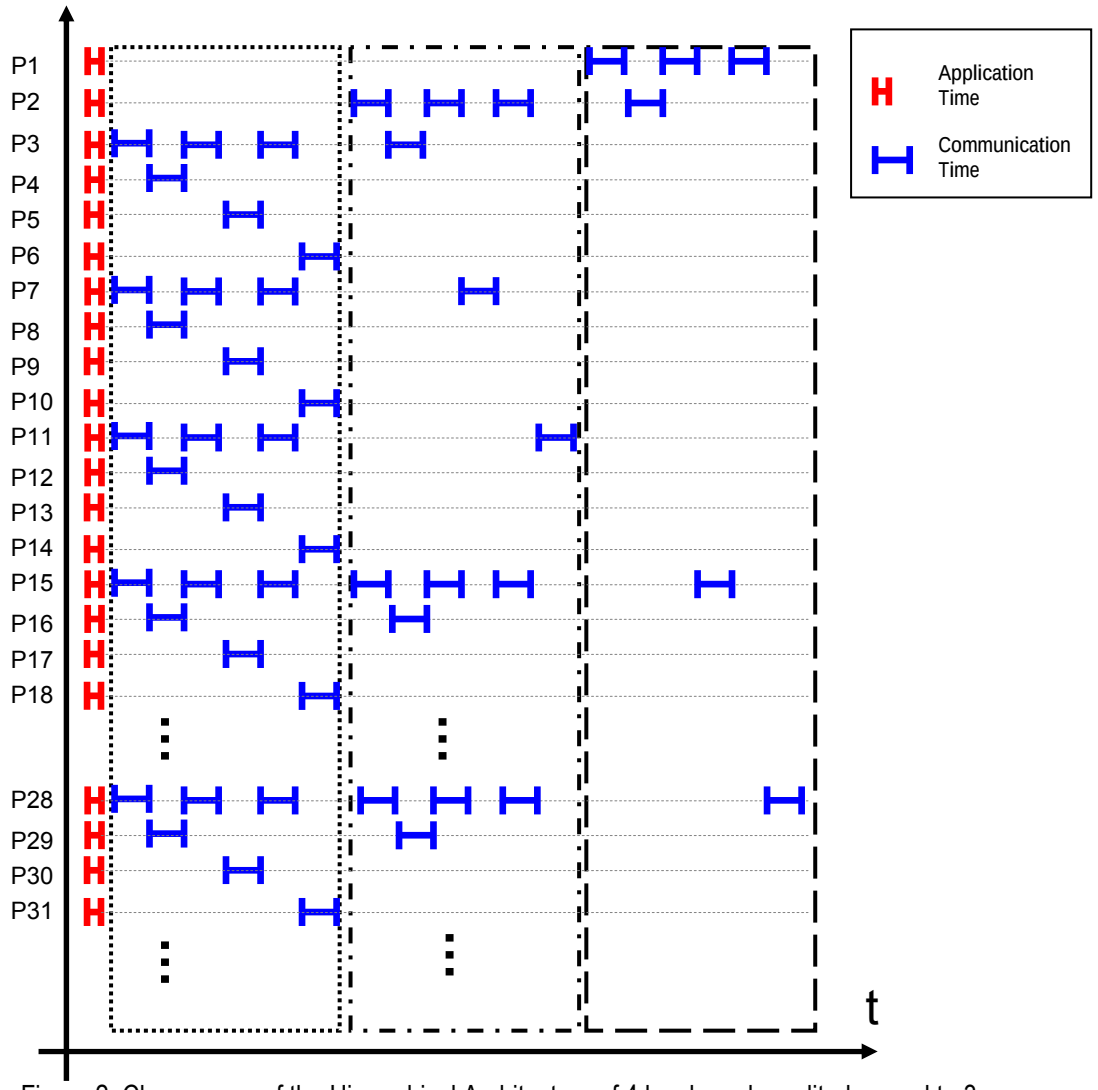


Figure 2. Chronogram of the Hierarchical Architecture of 4 levels and amplitude equal to 3.

Considering the hierarchical distribution of processors, the pillars of this model are:

a. Membranes distribution as in [Tejedor, 2007].
b. Proxy for processor as in [Tejedor, 2007].
c. Balanced tree topology of processors. Benefit obtained from this interconnection topology among processors is that the number of total external communications is minimized because proxys only exchange information with their direct descendants so, total number of external communications is *2(P-1)*, where

$$P = \frac{A^N - 1}{A - 1} \tag{4}$$

d. Token passing in the communication. A sequential order of communication is established for each processor in the same subtree; this way, there can not be more than proxy trying to transmit at the same time in the same subtree which is in. But, sequential external communications of a subtree are carried out in parallel with the ones of any other subtree of the same level. Last, established order for different levels is bottom-upm, i. e., no subtree of a given level begins its communications until every subtree of lower levels have finished.

This communication policy avoids collisions and network congestion, but additionally permits to be parallelized the 2(P-1) external communications so, the longest external communication sequence in each evolution step will be:

$$2(A-1)(N-1) \tag{5}$$

Hence, in this hierarchical architecture *K* membranes have been located in each processor. At the worst, the application of the rules in each one of these membranes will be made sequentially in each processor. Therefore, the required time to carry out the application of the rules of *M* membranes will be:

$$K\,T_{apl} \tag{6}$$

From (1), (4) and (6) the required time to carry out the application of the rules of *M* membranes will be:

$$M\,\frac{(A-1)}{A^N-1}T_{apl} \tag{7}$$

On the other hand, from (5) it is obtained the required time to carry out the communication among processors of the architecture:

$$2(A-1)(N-1)T_{com} \tag{8}$$

Therefore, from (7) and (8) the required time to perform a complete evolution step will be:

$$T = M\,\frac{(A-1)}{A^N-1}T_{apl} + 2T_{com}(N-1)(A-1) \tag{9}$$

Once the required time to perform an evolution step is known, we can determine the number of levels (L_{opt}) and the amplitude (A_{opt}) of the architecture in order to minimize this time:

$$A_{opt} = 2 \tag{10}$$

$$L_{opt} = \frac{\ln\left(\sqrt{T_{apl}\,\frac{M}{T_{com}}\ln(2)}\cdot\sqrt{T_{apl}\,\frac{M}{T_{com}}\ln(2)+8}+T_{apl}\,\frac{M}{T_{com}}\ln(2)+4\right)}{\ln(2)}-2 \tag{11}$$

From (9) and (10) the minimum time required to perform an evolution step is:

$$T_{\min} = \frac{M}{2^{L_{opt}}-1}T_{apl} + 2T_{com}(L_{opt}-1) \tag{12}$$

And, from (4) and (10) the number of processors necessary to run the P system minimizing the necessary time to carry out an evolution step will be:

$$P_{opt} = 2^{L_{opt}}-1 \tag{13}$$

Comparative Analysis

In this section, we present an empirical analysis comparing proposed architectures in [Tejedor, 2007] with the hierarchical architecture proposed here.

Figure 3 shows the number of processors of both architectures to reach their respective optimum times for an evolution step. As it can be seen, hierarchical architecture have a bigger number of processors than previous work. Also, the growing slope becomes steeper as the number of membranes of the P system is growing. This way, hierarchical architecture reaches a better parallelism degree in proportion to a bigger number of processors in the architecture. This fact increases the parallel application of evolution rules and the internal communication among membranes allocated at the same processor.

Consequently, the bigger parallelization degree of our architecture and external communications parallelization between subtrees of same level obtains smaller minimum times per evolution step. Figure 4 shows resulting times for both architectures as the number of membranes of the P system grow up.

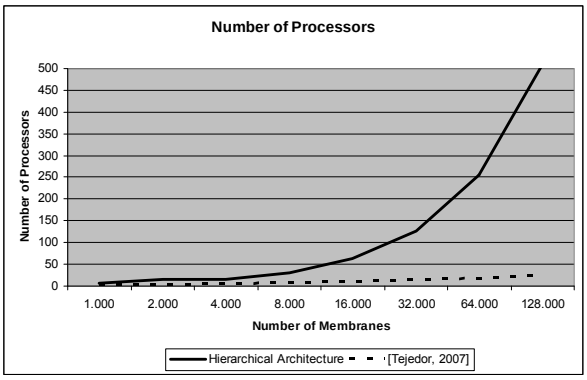


Figure 3. Number of processors to reach optimum times per evolution step among membranes in both architectures.

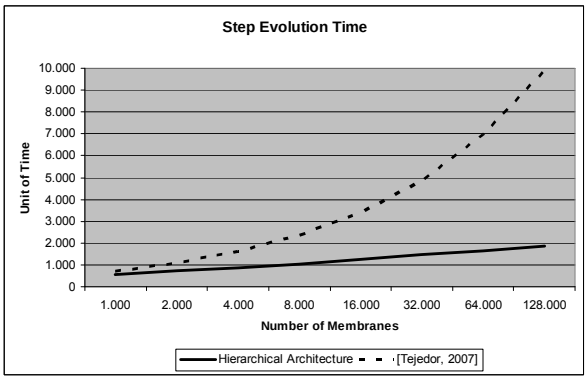


Figure 4. Optimum times per evolution step in both architectures.

Conclusions

In this paper a hierarchical architecture of communications to implement P system has been introduced. This architecture is based on the location of several membranes at the same processor, the use of proxys for communicating processors placed in a balanced tree topology and token passing in the communication.

This solution, just like previous architectures, avoids communication collisions, reduces the number and length of the external communications, but permits for the first time the parallelization of external communications and increases drastically the application rules and internal communications parallelization degree. All this, allows us to obtain a better step evolution time than any other suggested architectures and is closer to the massively parallelism character inherent to the membranes computer model.

Bibliography

[Păun, 1998] Gh.Păun. Computing with membranes. Journal of Computer and System Sciences, 61 (2000), and Turku Center for Computer Science-TUCS Report No 208, 1998.

[Tejedor, 2007] A. Tejedor, L. Fernandez, F. Arroyo, G. Bravo, An architecture for attacking the bottleneck communication in P Systems. In: M. Sugisaka, H. Tanaka (eds.), Proceedings of the 12th Int. Symposium on Artificial Life and Robotics, Jan 25-27, 2007, Beppu, Oita, Japan, 500-505.

[Ciobanu, 2004] G.Ciobanu, W.Guo. P Systems Running on a Cluster of Computers. Workshop on Membrane Computing (Gh. Păun, G. Rozenberg, A. Salomaa Eds.), LNCS 2933, Springer, 123-139, 2004.

[Syropoulos, 2003] A. Syropoulos, E.G. Mamatas, P.C. Allilomes, K.T. Sotiriades, A distributed simulation of P systems, A. Alhazov, C. Martin-Vide and Gh. Păun (Editors): Preproceedings of the Workshop on Membrane Computing; Tarragona, July 17-22 2003, 455-460.

Authors' Information

Ginés Bravo García – Natural Computing Group of Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: gines@eui.upm.es

Luis Fernández Muñoz – Natural Computing Group of Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: setillo@eui.upm.es

Fernando Arroyo Montoro – Natural Computing Group of Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: farroyo@eui.upm.es

Juan Alberto Frutos Velasco – Natural Computing Group of Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: jafrutos@eui.upm.es

STATIC ANALYSIS OF USEFULNESS STATES IN TRANSITION P SYSTEMS

Juan Alberto Frutos, Luis Fernandez, Fernando Arroyo, Gines Bravo

Abstract: Transition P Systems are a parallel and distributed computational model based on the notion of the cellular membrane structure. Each membrane determines a region that encloses a multiset of objects and evolution rules. Transition P Systems evolve through transitions between two consecutive configurations that are determined by the membrane structure and multisets present inside membranes. Moreover, transitions between two consecutive configurations are provided by an exhaustive non-deterministic and parallel application of evolution rules. But, to establish the rules to be applied, it is required the previous calculation of useful, applicable and active rules. Hence, computation of useful evolution rules is critical for the whole evolution process efficiency, because it is performed in parallel inside each membrane in every evolution step. This work defines usefulness states through an exhaustive analysis of the P system for every membrane and for every possible configuration of the membrane structure during the computation. Moreover, this analysis can be done in a static way; therefore membranes only have to check their usefulness states to obtain their set of useful rules during execution.

Keywords: Evolution Rules, Usefulness States, Transition P System, Sequential Machines, Static Analysis

ACM Classification Keywords: F.1.1 Computation by abstract devices – Models of computation. D.1.m Miscellaneous – Natural Computing

Introduction

Membrane Computing was introduced by Gh. Păun in [Păun, 1998], as a new branch of natural computing, inspired on living cells. Membrane systems establish a formal framework in which a simplified model of cells is considered a computational device. Starting from a basic model, Transition P systems, many different variant have been considered; and many of them have been demonstrated to be, in power, equivalent to the Turing Machine. An overview of this model is described in the next section.

Nowadays, a challenge for researchers of these kinds of devices is to get real implementations of membrane systems with a high degree of parallelism. Accordingly with this fact, there are some published works related to parallel implementation of membrane systems [Ciobanu, 2004], [Syropoulos, 2003] and [Tejedor, 2007].

In [Tejedor, 2007] set up two different phases in the inner dynamic of the evolution step: first phase is related to inner application of evolution rules inside membranes; second phase is related to communication among membranes in the systems. Then it is computed the total time the system spend during the evolution step, and what is important to note is the fact that reducing the time membranes spend in the application phase, the system gets an important gain in the total time it needs for the evolution step. The work presents in this paper is to improve the first phase –application of evolution rules inside membranes- getting useful rules in a faster way. In order to do it, it is introduced the concept of *usefulness states* of membranes in Transition P systems. The main idea is to carry out a static analysis of the P system in order to obtain all usefulness states and transitions between states in each membrane. During execution, membranes will obtain the set of useful evolution rules directly from their usefulness states.

This paper is structures as follows: first Transition P systems are formally defined. Second, usefulness states associated to membranes of Transition P systems with rules able to dissolve membranes are established. Third, the inhibition capability in P systems is incorporated. Fourth, a way for encoding usefulness states is introduced in order to reduce the needed space for implementing. Finally, conclusions are presented.

Transition P Systems

Formally, a transition P system of degree *m* is a construct of the form

$$\Pi = (O, \mu, \omega_1, \ldots, \omega_m, (R_1, \rho_1), \ldots, (R_m, \rho_m), i_0) \text{ , where:}$$

- *O* is the alphabet of objects
- μ is a membrane structure, consisting of *m* membranes, labelled with *1,2,...., m*. It is a hierarchically arranged set of membranes, contained in a distinguished external membrane, called skin membrane.

Several membranes can be placed inside a parent membrane; and finally, a membrane without any other membrane inside is said to be elementary.

- $\omega_i \mid 1<=i<=m$ are strings over O, representing multisets of objects placed inside the membrane with label *i*.
- $R_i \mid 1<=i<=m$ are finite sets of evolution rules associated to the membrane with label *i*. Rules have the form $u \rightarrow v$, $u \rightarrow v\,\delta$ or $u \rightarrow v\,\tau$, with $u \in O^+$ and $v \in (O^+ \times TAR)^*$, where $TAR=\{here, out\} \cup \{in_j \mid 1 <= i <= m\}$. Symbol δ represents membrane dissolution, while symbol τ represents membrane inhibition. ρ_i, $1 <= i <= m$, are priority relations defined over R_i, the set of rules of membrane *i*.
- i_0 represents the label of the membrane considered as output membrane.

The initial configuration of a P system is given by specifying the membrane structure and the multisets of objects placed inside membranes. $C = (\mu, \omega_1, \ldots, \omega_m)$. A transition takes place by application of evolution rules inside each membrane in the system, in a non-deterministic and maximally parallel manner. This implies that every object in the system able to evolve by the application of one evolution rule must evolve and rules are applied in a non-deterministic way. A computation is defined as a sequence of transitions between system configurations in which the final configuration has no objects able to evolve at any membrane of the system.

Figure 1 shows an example of transition P system, although only multiset and rules associated to membrane 1 are represented.

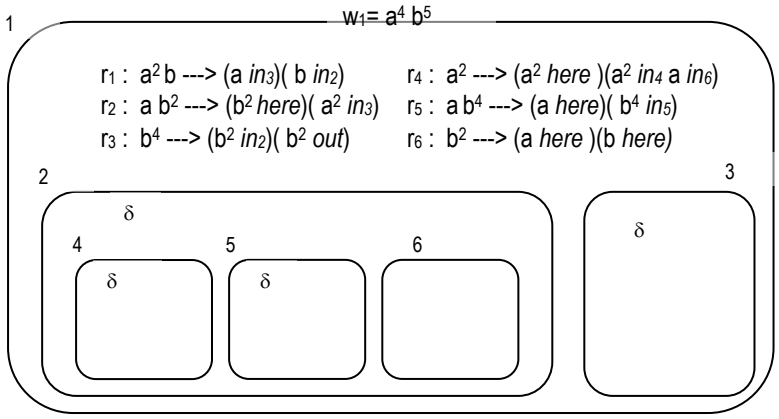


Figure 1: Transition P System

Usefulness states in transition P systems with membrane dissolving capability

Transition P systems with membrane dissolving capability are characterized by $OP_m(\alpha,tar,\delta,pri)$. This notation denotes the class of P systems with simple objects, priorities and dissolving capability. In this class of P systems, rules have the capability for dissolving membranes in the systems and, hence, they can modify the membrane structure of the P system during execution. Evolution rules in these systems are of the form $u \rightarrow v$ *or* $u \rightarrow v\delta$. In a different way, $r = (u, v, \xi), where\ \xi \in \{\delta, \lambda\}$.

Evolution rules able to be applied at any evolution step of the P system must accomplish three requisites: useful, applicable and active. A rule is *useful* in an evolution step if all targets are adjacent and not dissolved. In membrane 1 of the Figure 1, evolution rule r_4 is not useful in the initial configuration, but if membrane 2 is dissolved then membrane 4 and 6 become adjacent, and rule r_4 useful. On the other hand, a rule is *applicable* if its antecedent is included in the multiset of the membrane. Finally, a rule is *active* if there is no other applicable rule with higher priority.

The main goal of this work is to reduce the time of getting useful rules, avoiding communication as much as possible. The proposed solution is to define the membrane context associated to membranes and configurations in the P system.

Definition: The *membrane context* in a time is the set of children membranes to which rules can send objects in the current membrane structure of the system. These membranes are adjacent to the current one.

The basic idea is the following: every membrane in the P system has to know its context at every time. When a membrane is dissolved, then it has to report the dissolution to its father, and the latter will update its context.

Definition: a *usefulness state* q_i^j in a membrane *j* is a valid context in that membrane, $C(q_i^j)$. A context in a membrane is valid if it could be reached in any configuration of the system.

The target of this work is to find out statically all valid usefulness states at any membrane of a P System, the useful rules associated to each usefulness state, and transitions between states when membranes are dissolved.

Definition: Let $Child\_Of(j) = \{i \mid 1 <= i <= m;\ i \text{ is a child of } j \text{ en } \mu\}$, that is, all membrane *j* children in μ.

Definition: Let Q^j the set of *Usefulness states* associated to the membrane labelled with *j* in the P system Π, defined as follows:

1. if the membrane *j* is an elementary membrane: $Q^j = \{q_0^j\}$, *where* $C(q_0^j) = \{\varnothing\}$
2. if the membrane *j* is not an elementary membrane:

$$Q^j = \underset{i \in Child\_Of(j)}{X} Q'^{\,i}, \text{ where } Q'^{\,i} = \begin{cases} \{q_N^i\} \text{ if membrane } i \text{ cannot be dissolved} \\ \{q_N^i\} \cup Q^i \text{ if membrane } i \text{ can be dissolved} \end{cases}$$

q_N^i is a state representing that membrane *i* is not dissolved, therefore the context in q_N^i is $C(q_N^i) = \{i\}$.

Context for each one of the states belonging to the *Cartesian product* is obtained by the union of contexts which configure the corresponding state. $C((q_{s_1}^{i_1},..,q_{s_n}^{i_n})) = \bigcup_{i_k \in Child\_Of(j)} C(q_{s_k}^{i_k})$.

Considering the P system depicted in Figure 1, only evolution rules associated to membrane 1 are shown. Other membranes only show if there is any rule which can dissolve them; hence membranes with labels 2, 3, 4 and 5 can be dissolved during execution of the system. In order to determine *Usefulness states* per membrane, we shall start from inside to outside of the membrane system; that is, from elementary membranes to the skin membrane.

It seems to be clear that elementary membranes cannot have more than one state, with null context. Therefore, $Q^3 = \{q_0^3\}$, $Q^4 = \{q_0^4\}$, $Q^5 = \{q_0^5\}$ and $Q^6 = \{q_0^6\}$. Each one of these states has context $\{\varnothing\}$.

For membrane 2:

$$Q^2 = Q'^4 \times Q'^5 \times Q'^6$$

$$Q'^4 = \{q_N^4\} \cup \{q_0^4\} \quad Contexts = \{\{4\},\{\varnothing\}\}$$

$$Q'^5 = \{q_N^5\} \cup \{q_0^5\} \quad Contexts = \{\{5\},\{\varnothing\}\}$$

$$Q'^6 = \{q_N^6\} \quad Contexts = \{\{6\}\}$$

$$Q^2 = \{\overbrace{(q_N^4,q_N^5,q_N^6)}^{q_0^2},\overbrace{(q_N^4,q_0^5,q_N^6)}^{q_1^2},\overbrace{(q_0^4,q_N^5,q_N^6)}^{q_2^2},\overbrace{(q_0^4,q_0^5,q_N^6)}^{q_3^2}\} \quad Contexts = \{\{4,5,6\},\{4,6\},\{5,6\},\{6\}\}$$

And finally, for membrane 1:

$$Q^1 = Q'^2 \times Q'^3$$

$$Q'^2 = \{q_N^2\} \cup \{q_0^2,q_1^2,q_2^2,q_3^2\} \quad Contexts = \{\{2\},\{4,5,6\},\{4,6\},\{5,6\},\{6\}\}$$

$$Q'^3 = \{q_N^3\} \cup \{q_0^3\} \quad Contexts = \{\{3\},\{\varnothing\}\}$$

$$Q^1 = \{\overbrace{(q_N^2,q_N^3)}^{q_0^1},\overbrace{(q_N^2,q_0^3)}^{q_1^1},\overbrace{(q_0^2,q_N^3)}^{q_2^1},\overbrace{(q_0^2,q_0^3)}^{q_3^1},\overbrace{(q_1^2,q_N^3)}^{q_4^1},\overbrace{(q_1^2,q_0^3)}^{q_5^1},\overbrace{(q_2^2,q_N^3)}^{q_6^1},\overbrace{(q_2^2,q_0^3)}^{q_7^1},\overbrace{(q_3^2,q_N^3)}^{q_8^1},\overbrace{(q_3^2,q_0^3)}^{q_9^1}\}$$

$$Contexts = \{\{2,3\},\{2\},\{4,5,6,3\},\{4,5,6\},\{4,6,3\},\{4,6\},\{5,6,3\},\{5,6\},\{6,3\},\{6\}\}$$

Useful rules associated to usefulness states

Every Usefulness state is characterized by its context, that is, the set of children membranes directly enclosed in the original membrane. Hence, the context or state determines the set of useful rules in the membrane. Moreover, what is important to note is that the set of usefulness states, contexts and, hence, the set of evolution rules for each one of the membranes and possible configuration of the system can be established in a static analysis.

Lemma: An evolution rule $r = (u, v\xi), where\ \xi \in \{\delta, \lambda\}$ is useful in q_i^j if and only if $\forall\ TAR\ in_k \in v, k \in C(q_i^j)$.

Considering the previous P system Π for membrane 1, the table 1 shows the whole set of usefulness states – contexts and their corresponding sets of useful evolution rules accordingly to the states.

Usefulness states	Useful Rules
q_0^1 {2, 3}	r_1, r_2, r_3, r_6
q_1^1 {2}	r_3, r_6
q_2^1 {4, 5, 6, 3}	r_2, r_4, r_5, r_6
q_3^1 {4, 5, 6}	r_4, r_5, r_6
q_4^1 {4, 6, 3}	r_2, r_4, r_6
q_5^1 {4, 6}	r_4, r_6
q_6^1 {5, 6, 3}	r_2, r_5, r_6
q_7^1 {5, 6}	r_5, r_6
q_8^1 {6, 3}	r_2,, r_6
q_9^1 {6}	r_6

Table 1: Useful Evolution Rules associated to Usefulness States for Membrane 1

transitions between usefulness states

Definition: Let $Child\_D(j) = \{i \in Child(j) \wedge \exists r = (u,v,\delta) \in R_j\}$, be the set of child membranes to membrane *j* that can be dissolved.

Definition: Let $TC\_D(j) = Child\_D(j) \bigcup_{i \in Child\_D(j)} TC\_D(i)$, be the total context for membrane *j*, including only those membranes that can be dissolved. By total context is understood those membrane that eventually can become children of membrane *j*.

A transition between two *usefulness states* in a membrane is produced when a child membrane is dissolved. In this case, father membrane is affected and its usefulness state must change. The way for representing this behaviour is through a Moore's Sequential Machine in every membrane labelled with *j*.

$$MS^j = \left(\sum\nolimits_I^j, \sum\nolimits_O^j, Q^j, q_0^j, g^j, f^j\right), \text{ where:}$$

- Input alphabet: $\sum\nolimits_I^j = \{\delta(i, q_s^i) \mid i \in TC\_D(j), q_s^i \in Q^i\}$, the sequential machine will transits when a child membrane is dissolved. Child membrane must send to membrane *j* that is dissolved and its usefulness state because the context of the membrane child will pass to be part of the parent context.
- Output alphabet: $\sum\nolimits_O^j = \{r_k \mid r_k \in R_j\}$, the set of useful rules in membrane *j*.
- Set of states: $Q^j = \{(q_{s_1}^{i_1}, .., q_{s_n}^{i_n}) \mid i_k \in Child\_Of(j), q_{s_k}^{i_k} \in Q'^{i_k}\}$, the set of usefulness states of membrane *j*.
- Initial state: $(q_N^{i_1}, .., q_N^{i_n}) \mid i_k \in Child\_Of(j)$, that is, the state in which every child membrane is not dissolved.
- Output function: $g^j : Q \to \mathcal{P}(R_j)$. the function that assigns a set of useful rules to each one of the *usefulness state* of the membrane *j*; as it was shown in table 1.
- Transition function: $f^j : Q \times \sum_I^j \to Q$. the function provides the new usefulness state to transit given the current one and the dissolution of a child membrane. This function is defined as follows: $\forall i_k \in Child\_D(j)$

1) If i_K is dissolved, $i_k : f^j((q_{s_1}^{i_1}, ..., q_N^{i_k}, ..., q_{s_n}^{i_n}), \delta(i_k, q_s^{i_k})) = (q_{s_1}^{i_1}, ..., q_s^{i_k}, ..., q_{s_n}^{i_n})$.
2) If membrane m is dissolved being child of *j* and $m \in TC\_D(i_k)$:

$$f^j((q_{s_1}^{i_1}, ..., q_{s_k}^{i_k}, ..., q_{s_n}^{i_n}), \delta(m, q_s^m)) = (q_{s_1}^{i_1}, ..., q_p^{i_k}, ..., q_{s_n}^{i_n}) \quad where\ f^{i_k}(q_{s_k}^{i_k}, \delta(m, q_s^m)) = q_p^{i_k}$$

Hence, starting from states transition tables of children membranes, it will be obtained the transition table for membrane *j*. Of course, elementary membranes have not transition tables because of they have only one state. As an example, the transition function f^1 for membrane 1 of the P system of Figure 1 is depicted in table 2.

	$\delta(2,q_0^2)$	$\delta(2,q_1^2)$	$\delta(2,q_2^2)$	$\delta(2,q_3^2)$	$\delta(4,q_0^4)$	$\delta(5,q_0^5)$	$\delta(3,q_0^3)$
(q_N^2, q_N^3)	(q_0^2, q_N^3)	(q_1^2, q_N^3)	(q_2^2, q_N^3)	(q_3^2, q_N^3)	---	---	(q_N^2, q_0^3)
(q_N^2, q_0^3)	(q_0^2, q_0^3)	(q_1^2, q_0^3)	(q_2^2, q_0^3)	(q_3^2, q_0^3)	---	---	---
(q_0^2, q_N^3)	---	---	---	---	(q_2^2, q_N^3)	(q_1^2, q_N^3)	(q_0^2, q_0^3)
(q_0^2, q_0^3)	---	---	---	---	(q_2^2, q_0^3)	(q_1^2, q_0^3)	---
(q_1^2, q_N^3)	---	---	---	---	---	(q_3^2, q_N^3)	(q_1^2, q_0^3)
(q_1^2, q_0^3)	---	---	---	---	---	(q_3^2, q_0^3)	---

(q_2^2, q_N^3)	---	---	---	---	(q_3^2, q_N^3)	---	(q_2^2, q_0^3)
(q_2^2, q_0^3)	---	---	---	---	(q_3^2, q_0^3)	---	---
(q_3^2, q_N^3)	---	---	---	---	---	---	(q_3^2, q_0^3)
(q_3^2, q_0^3)	---	---	---	---	---	---	---

Table 2: Usefulness states transition function for membrane 1.

In Table 2 transitions for dissolutions of membranes 4 and 5 have been obtained from transition function of membrane 2 –shows in Table 3-, because they belong to the total context of membrane 2.

	$\delta(4, q_0^4)$	$\delta(5, q_0^5)$
q_0^2	q_2^2	q_1^2
q_1^2	---	q_3^2
q_2^2	q_3^2	---
q_3^2	---	---

Table 3: Usefulness states transition function for membrane 2.

As an example, if from the state (q_0^2, q_N^3) with context {4,5,6,3}, it is produced $\delta(4, q_0^4)$, then looking at transition table for membrane 2 from q_0^2 with $\delta(4, q_0^4)$, the result is q_2^2, and then the corresponding transition is to (q_2^2, q_N^3) with context {5,6,3}.

Finally, it can be changed the notation for representing usefulness states, in this case, they are numbering in a correlative manner starting from 0. That is, $\{q_0^1, q_1^1, q_2^1, q_3^1, q_4^1, q_5^1, q_6^1, q_7^1, q_8^1, q_9^1\}$, like in table 3 for membrane 2.

Usefulness states in transition P systems with Dissolution and Inhibition Capability.

Evolution rules in these systems are of the form, $u \to v\ \xi$, *where* $\xi \in \{\delta, \tau, \lambda\}$. Symbol τ indicates that after rule application membrane containing the rule will be not permeable to objects communication. This membrane will come back to be permeable to objects communication by the application of one evolution rule having the symbol δ. If during the application phase of evolution rules different rules having symbols δ and τ are applied, then membrane will not change its communication state.

Hence, it would be considered three different membrane states concerning to objects communication: Dissolved, Permeable and inhibited or impermeable. These three states and their transition graph are depicted in Figure 2

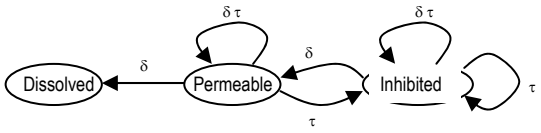


Figure 2

Inhibition capability modifies the previous study for *usefulness states* where only dissolution was allowed. Application of rules having the τ symbol in a membrane could modify its capability for accepting objects coming from outside. Hence, this fact modifies the context of the parent membrane, because rules sending objects to a membrane in the inhibited state are not useful.

Definition: Let Q^j the set of *Usefulness states* associated to the membrane labelled with *j*, defined as follows:

1. if the membrane *j* is an elementary membrane: $Q^j = \{q_0^j\}$, *where* $C(q_0^j) = \{\varnothing\}$
2. if the membrane *j* is not an elementary membrane: $Q^j = \underset{i \in Child\_Of(j)}{X} Q'^{\,i}$, where

$$Q'^{\,i} = \begin{cases} \{q_N^i\}\ \textit{if membrane i can be neither dissolved nor inhibited} \\ \{q_N^i, q_I^i\}\ \textit{if membrane i can be inhibited, but not dissolved} \\ \{q_N^i\} \cup Q^i\ \textit{if membrane i can be dissolved, but not inhibited} \\ \{q_N^i, q_I^i\} \cup Q^i\ \textit{if membrane i can be inhibited and dissolved} \end{cases}$$

q_N^i represents the permeable state fo the membrane i, therefore $C(q_N^i) = \{i\}$.

q_I^i represents the inhibited state of the membrane *i*, therefore $C(q_I^i) = \varnothing$.

Useful rules associated to usefulness states

In order to determine which evolution rules are useful in a determined membrane and evolution step, now it is needed to assure not only that evolution rules targets of type in_k are all of then in the membrane context, but also current membrane must be permeable if target of type *out* is included. Hence, it is needed to consider the *usefulness state* and permeability state of the membrane; and then, it could be possible to abroad the static analysis of usefulness states for P systems with membrane dissolution and permeability control.

Lemma: An evolution rule $u \to v\,\xi$, *where* $\xi \in \{\delta, \tau, \lambda\}$ is useful in q_i^j *and* q_{perm}^j if and only if $\forall\, TAR\ in_k \in v, k \in C(q_i) \wedge Si\ \exists\, TAR\ out \in v, q_{perm}^j = Permeable$

Transitions between usefulness states

Definition: Let $Child\_I(j) = \{i \in Child\_Of(j) \wedge \exists r = (u, v, \tau) \in R_j\}$ be the set of children membranes of membrane *j* that can be inhibited.

Definition: Let $TC\_I(j) = Child\_I(j) \bigcup_{i \in Child\_D(j)} TC\_I(i)$, be the membrane *j* total context considering only those children membranes that can be inhibited.

In these systems, transitions are not only produced by membranes dissolution (δ), but also with membranes inhibition (τ) and come back permeable ($\neg\tau$). Therefore, the alphabet for the sequential states machines is:

$$\sum\nolimits_I^j = \{\delta(i, q_s^i) \mid i \in TC\_D(j), q_s^i \in Q^i\} \cup \{\tau\, i \mid i \in TC\_I(j)\} \cup \{\neg\tau\, i \mid i \in TC\_D(j) \cap TC\_D(j)\}$$

And the transition function is:

$\forall i_k \in Child\_Of(j)$

If i_k is dissolved: $f^j((q_{s_1}^{i_1}, ..., q_N^{i_k}, ..., q_{s_n}^{i_n}), \delta(i_k, q_s^{i_k})) = (q_{s_1}^{i_1}, ..., q_s^{i_k}, ..., q_{s_n}^{i_n})$

if *m* is dissolved being child of *j* and $m \in TC\_D(i_k)$: $f^j((q_{s_1}^{i_1}, ..., q_{s_k}^{i_k}, ..., q_{s_n}^{i_n}), \delta(m, q_s^m)) = (q_{s_1}^{i_1}, ..., q_p^{i_k}, ..., q_{s_n}^{i_n})$

$$where\ f^{i_k}(q_{s_k}^{i_k}, \delta(m, q_s^m)) = q_p^{i_k}$$

If i_k is inhibited: $f^j((q_{s_1}^{i_1}, ..., q_N^{i_k}, ..., q_{s_n}^{i_n}), \tau\, i_k) = (q_{s_1}^{i_1}, ..., q_I^{i_k}, ..., q_{s_n}^{i_n})$

If *m* is inhibited being child of *j* and $m \in TC\_I(i_k)$: $f^j((q_{s_1}^{i_1}, ..., q_{s_k}^{i_k}, ..., q_{s_n}^{i_n}), \tau\, m) = (q_{s_1}^{i_1}, ..., q_p^{i_k}, ..., q_{s_n}^{i_n})$

$$where\ f^{i_k}(q_{s_k}^{i_k}, \tau\, m) = q_p^{i_k}$$

If i_k comes back to be permeable: $f^j((q_{s_1}^{i_1}, ..., q_I^{i_k}, ..., q_{s_n}^{i_n}), \neg\tau\, i_k) = (q_{s_1}^{i_1}, ..., q_N^{i_k}, ..., q_{s_n}^{i_n})$

if *m* comes back to be permeable being child of *j* and $m \in TC\_D(i_k) \cap TC\_I(i_k)$:

$$f^j((q_{s_1}^{i_1}, ..., q_{s_k}^{i_k}, ..., q_{s_n}^{i_n}), \neg\tau\, m) = (q_{s_1}^{i_1}, ..., q_p^{i_k}, ..., q_{s_n}^{i_n})\ where\ f^{i_k}(q_{s_k}^{i_k}, \neg\tau\, m) = q_p^{i_k}$$

Encoding usefulness states

The main problem when usefulness states are encoded in a determined Hardware/Software architecture could be the size of transition states tables used for representing usefulness states transition functions in membranes. This is the reason why in this paper is proposed a way for encoding usefulness states with the purpose of making transition without using usefulness states transition tables.

Definition:

Let $TC(j) = Child\_Of(j) \bigcup_{i \in Child\_D(j)} TC(i)$, the total context for membrane *j*, independently of dissolving or inhibition.

The appearing membranes order in $TC(j)$, is normalized going down into the sub-tree of μ starting in membrane *j* in depth and in pre-order. And they are represented in this order in the *Normalized Total Context* of membrane *j*.

Definition:

Let $TC_{Normal}(j) = (i_1, TC_{Normal}(i_1), ..., i_n, TC_{Normal}(i_n))$ *where* $i_k \in Child\_Of(j)$ *from left to right in* μ

Each one of the usefulness states of membrane *j*, q_i^j is enconded on $TC_{Normal}(j)$ depending on its context, $C(q_i^j)$, with binary logic. The value 1 set out that membrane *k* is present in $C(q_i^j)$, while value 0 will represents

that membrane *k* is not in $C(q_i^j)$. As an example, for membrane 1 of the P system depicted in Figure 1, it is obtained the total context $TC_{Normal}(1) = (2,4,5,6,3)$, and the usefulness states enconded are represented in table 4.

If $q^j(t) = (i_1,....,i_k,.....,i_n)$ *encoded by* $TC_{Normal}(j)$ is the usefulness state of membrane *j* at time *t*, the transitional logic will be the following:

1. If the child membrane of *j*, i_k, at time *t* is inhibited: $q^j(t+1) = (i_1,....,0,.....,i_n)$
2. If the child membrane of *j*, i_k, at time *t* comes back to be permeable: $q^j(t+1) = (i_1,....,1,.....,i_n)$
3. If the child membrane of *j*, i_k, at time *t* is dissolved, it has to send its usefulness state $q^{i_k}(t+1)$, encoded by its normalized total context, $TC_{Normal}(i_k)$. It can be considered in a deeper sight the usefulness state for membrane *j* as $q^j(t) = (i_1,....,i_k,TC(i_k),.....,i_n)$ and the transition is $q^j(t+1) = (i_1,....,0,q^{i_k}(t+1),.....,i_n)$

Usefulness states	Encoding
q_0^1 {2, 3}	10001
q_1^1 {2}	10000
q_2^1 {4, 5, 6, 3}	01111
q_3^1 {4, 5, 6}	01110
q_4^1 {4, 6, 3}	01011
q_5^1 {4, 6}	01010
q_6^1 {5, 6, 3}	00111
q_7^1 {5, 6}	00110
q_8^1 {6, 3}	00011
q_9^1 {6}	00010

Table 4: Encoding of usefulnes states

In the proposed example, if membrane 1 is in usefulness state $q^1(t) = (10001)$ and membrane 2 is dissolved in $q^2(t) = (101)$ encoded by its normalized total context $TC_{Normal}(2) = (4,5,6)$, it is obtained the transition $q^1(t+1) = (01011)$. This is the transition of table 2 $f^1((q_N^2,q_N^3),\delta(2,q_1^2)) = (q_1^2,q_N^3)$ without making use of table.

Conclusion

This paper presents the study of usefulness states associated to membranes of Transition P system. The aim of the work developed here is to reduce the evolution rules application time. In order to get the necessary efficiency in the application phase of rules, the analysis of usefulness states can be done in a static manner, and this implies an important reduction in time needed for evolution steps in the system. Moreover, not only usefulness states are defined here, but also the logic of transition between them. Each one of the usefulness states is associated to its own set of useful rules, and in this way there is no computation needed to obtain them because the computation of usefulness states and context is done before starting system execution or simulation.

Bibliography

[Ciobanu 2004] G.Ciobanu, G.Wenyuan, "A P System runnning on a cluster of computers", Proceedings of Membrane Computing. International Workshop, Tarragona (Spain). Lecture Notes in Computer Science, vol 2933, 123-150.

[Păun, 1998] Gh.Păun, "Computing with Membranes", Journal of Computer and System Sciences, 61(2000), and Turku Center of Computer Science-TUCS Report nº 208, 1998.

[Syropoulos 2003] A. Syropoulos, E.G. Mamatas, P.C. Allilomes, K.T. Sotiriades, "A distributed simulation of P systems". Preproceedings of the Workshop on Membrane Computing (A. Alhazov, C.Martin-Vide and Gh.Păun, eds); Tarragona, vol July 17-22 (2003), 455-460.

[Tejedor, 2007] J.Tejedor, L.Fernández, F.Arroyo, G.Bravo, *An architecture for attacking the bottleneck communication in P systems*. In: M. Sugisaka, H. Tanaka (eds.), Proceedings of the 12th Int. Symposium on Artificial Life and Robotics, Jan 25-27, 2007, Beppu, Oita, Japan, 500-505.

Authors' Information

Juan Alberto de Frutos – e-mail: jafrutos@eui.upm.es

Luis Fernández – e-mail: setillo@eui.upm.es

Fernando Arroyo – e-mail: farroyo@eui.upm.es

Gines Bravo – e-mail: gines@eui.upm.es

Dpto. Lenguajes, Proyectos y Sistemas Informáticos (LPSI) de la Escuela Universitaria de Informática (EUI) de la Universidad Politécnica de Madrid (UPM); Ctra. Valencia, km. 7, 28031 Madrid (Spain).

DELIMITED MASSIVELY PARALLEL ALGORITHM BASED ON RULES ELIMINATION FOR APPLICATION OF ACTIVE RULES IN TRANSITION P SYSTEMS

Francisco Javier Gil, Luis Fernández, Fernando Arroyo, Jorge Tejedor

Abstract: In the field of Transition P systems implementation, it has been determined that it is very important to determine in advance how long takes evolution rules application in membranes. Moreover, to have time estimations of rules application in membranes makes possible to take important decisions related to hardware / software architectures design.

The work presented here introduces an algorithm for applying active evolution rules in Transition P systems, which is based on active rules elimination. The algorithm complies the requisites of being nondeterministic, massively parallel, and what is more important, it is time delimited because it is only dependant on the number of membrane evolution rules.

Keywords: Natural computing, Membrane computing, Transition P systems, rules application algorithms

ACM Classification Keywords: D.1.m Miscellaneous – Natural Computing

Introduction

Transition P systems are a distributed parallel computational model introduced by Gheorghe Păun based on basic features of biological membranes and the observation of biochemical processes [Păun, 1998]. In this model, membrane contains objects multisets, which evolve according to given evolution rules. Applying the later ones in a nondeterministic maximally parallel way the system changes from a configuration to another one making a computation. This model has become, during last years, an influential framework for developing new ideas and investigations in theoretical computation. “P systems with simple ingredients (number of membranes, forms and sizes of rules, controls of using the rules) are Turing complete” [Păun, 2005]. Moreover, P systems are a class of distributed, massively parallel and non-deterministic systems. “As there do not exist, up to now, implementations in laboratories (neither in vitro or in vivo nor in any electronically medium), it seems natural to look for software tools that can be used as assistants that are able to simulate computations of P systems” [Ciobanu, 2006]. “An overview of membrane computing software can be found in literature, or tentative for hardware implementations, or even in local networks is enough to understand how difficult is to implement membrane systems on digital devices” [Păun, 2005].

In addition, Gheorghe Păun says that: “we avoid to plainly say that we have ‘implementations’ of P systems, because of the inherent non-determinism and the massive parallelism of the basic model, features which cannot be implemented, at least in principle, on the usual electronic computer -but which can be implemented on a dedicated, reconfigurable, hardware [...] or on a local network”. Thereby, there exists many P systems simulators in bibliography but “the next generation of simulators may be oriented to solve (at least partially) the problems of storage of information and massive parallelism by using parallel language programming or by using multiprocessor computers” [Ciobanu, 2006].

This work presents a time delimited massively parallel algorithm based on rules elimination for application of active rules in transition P systems. After this introduction, other related works appear, where the problem that is tried to solve is exposed. Later the massively parallel algorithm of application of rules appears developed, including the synchronization between processes and the analysis of its efficiency.

Related Work

J. Tejedor proposes a software architecture for attacking the bottleneck communication in P systems denominated “partially parallel evolution with partially parallel communications model” where several membranes are located in each processor, proxies are used to communicate with membranes located in different processors and a policy of access control to the network communications is mandatory [Tejedor, 2006]. This obtains a certain parallelism yet in the system and an acceptable operation in the communications. In addition, it establishes a set of equations that they allow to determine in the architecture the optimum number of processors needed, the

required time to execute an evolution step, the number of membranes to be located in each processor and the conditions to determine when it is best to use the distributed solution or the sequential one. Additionally it concludes that if the maximum application time used by the slowest membrane in applying its rules improves *N* times, the number of membranes that would be executed in a processor would be multiplied by $\sqrt{N}$, the number of required processors would be divided by the same factor, and the time required to perform an evolution step would improve approximately with the same $\sqrt{N}$ factor.

Therefore, to design software architectures it is precise to know the necessary time to execute an evolution step. For that reason, algorithms for evolution rules application that they can be executed in a delimited time are required, independently of the object multiset cardinality inside the membranes. Nevertheless, this information cannot be obtained with the present algorithms since its execution time depends on the cardinality of the objects multiset on which the evolution rules are applied.

In addition, Ciobanu presents several related papers about parallel implementation of P systems [Ciobanu 2002, 2004, 2006], in which "the rules are implemented as threads. At the initialization phase, one thread is created for each rule. Rule applications are performed in term of rounds" [Ciobanu, 2006]. Again, the author recognizes that: "since many rules are executing concurrently and they are sharing resources, a mutual exclusion algorithm is necessary to ensure integrity" [Ciobanu, 2004]. So, "when more than one rule can be applied in the same conditions, the simulator randomly picks one among the candidates" [Ciobanu, 2006]. Hence, processes will have pre-protocols and post-protocols for accessing to critical sections included into their code in order to work under mutual exclusion. Then, each evolution rule set associated to a membrane must access to the shared multiset of objects under mutual exclusion; but different sets of evolution rules associated to different membranes there are no competition among them because they are disjoint processes. Hence, some degree of parallelism is achieved spite of having a thread for each evolution rule. The implementation of evolution rules application will be concurrent inside membranes but not massively parallel.

On the other hand, L. Fernández proposes a massively parallel algorithm for evolution rules application [Fernández, 2006]. In this solution a process by each rule is generated and exist one more controller process that simulates the membrane containing the objects multiset. In a loop, each rule process proposes simultaneously a object multiset to be consumed and the membrane process determines if it is possible to apply the proposed multiset, until the proposal is correct. The algorithm execution finishes when there is no active rule. This last solution contains a high degree of parallelism, but its execution time is not delimited. Therefore, this algorithm is not appropriate to be used in the previously commented [Tejedor, 2006] software architecture.

Finally, in [Tejedor, 2007] is exposed an algorithm for application of evolution rules based on active rules elimination. In this algorithm, in each loop iteration all the rules -except the last one- are sequentially applied a random number of times. Next, the last active rule is applied the greater possible number of times, reason why it became inactive. This algorithm reaches a certain degree of parallelism, since one rule can be simultaneously several times applied in a single step. In this algorithm, the execution time depends on the number of rules, not of the objects multiset cardinality. In the experimental tests, this algorithm has obtained better execution times than the previously published sequential algorithms. This sequential solution is, of course, a minimal parallelism solution.

Delimited Massively Parallel Algorithm based on Rules Elimination for Application of Active Rules in Transition P Systems

Here we present a time delimited massively parallel algorithm for application of active rules. The initial input is a set of active evolution rules for the corresponding membrane -the rules are applicable and useful- and the initial membrane multiset of objects. The final results are the complete multiset of applied evolution rules and the obtained multiset of objects after rules application. In order to achieve this, we propose one process for each rule and one more controller process that simulate the membrane containing the multiset of objects.

The general idea is that each rule -except the last one- randomly proposes, in an independent manner, a multiset to be consumed from the membrane multiset (the obtained algorithm is nondeterministic due to this random proposal). If the addition of all the proposed multiset by rules is smaller than the membrane multiset, then the proposed multiset is subtracted from the membrane multiset. Next, the last active rule determines and applies its maximal applicability benchmark over the membrane multiset, subtracting the correspondent multiset from the membrane multiset. At this point, rules that are not applicable over the new membrane multiset finish their process execution –obviously, including the last active rule-. The resting active rules come back to the starting

point, and again, propose a new multiset to be consumed. This process is repeated until none rule is applicable over the membrane multiset.

This idea can be divided into seven phases:

Phase 1 *Membrane initialization*. A global probability for proposing multiset to be consumed by rules is initialized. One rule is able to consume with a determined probability; but if a rule is not allowed to consume then it will not propose multiset to be consumed until the next loop iteration.

This phase is performed only by the controller process, while rules are waiting to second phase.

Phase 2 *Evolution rules initialization*. Each rule -except the last active rule- determines its applicability benchmark to its maximal applicability benchmark over the membrane multiset. On the other hand, every rule is settled to the state in which rules can propose.

This phase is performed in parallel by every rule. The controller process -membrane- waits until phase 5.

Phase 3 *Multiset propositions*. Considering the global probability established by the membrane, each rule proposes in a randomly manner one multiset of objects to be consumed from the membrane multiset. The proposed rule multiset can be the empty multiset or the scalar product of its antecedent by a natural number chosen in a random manner in between 1 and its applicability benchmark.

This phase is performed in parallel for every evolution rule, except the last one.

Phase 4 *Sum of Multiset Proposals*. The addition of the proposed multisets by rules is performed two by two by neighborhood with respect to their number. For example, rule number 1 with rule number 2, rule number 3 with rule number 4, and so on. After finishing this step, the resulting multisets are added two by two again. For example, rule number 1 with rule number 3. And so on until reaching one single multiset. This way for adding multiset develops a binary tree of additions performed in parallel at each level of the tree.

This phase is performed in parallel for every rule.

Phase 5 *Proposal management and last active rule maximal application*. Membrane analyzes the proposed multiset by the rules. If the proposed multiset is valid (not empty and included in the membrane multiset), then the membrane subtract from its own multiset the proposed multiset from phase 4. Next, the membrane process determines and applies the maximal applicability benchmark of the last active rule over the membrane multiset, subtracting the correspondent multiset from the membrane multiset. Moreover, the membrane process indicates to the rule processes the executed operation. Finally, it initializes the information about its active evolution rules for the next loop in the algorithm.

This phase is performed only by the membrane process while rules wait until the phase 6.

Phase 6 *Checking rules halt*. Each one of the evolution rules accumulates the number of proposed application over the membrane multiset. Moreover, it computes its maximal applicability benchmark over the new resting membrane multiset for the next iteration and, if it is bigger than 0, they pass to the state in which rules can propose and indicate it into the active evolution rules data structure. Otherwise, they finish their execution.

This phase is performed in parallel by every evolution rule except into the access to the active evolution rules data structure. Membrane is waiting until phase 7.

Phase 7 *Checking membrane halt*. Membrane checks if there exists some active rule for the next loop and, in this case, it returns to establish the global probability to propose multisets by the rules. If so, it come back to phase 5 waiting the proposal management, otherwise it finishes the execution.

This phase is performed only by the membrane and the rules wait for coming back to phase 3 -if they are active for next loop- of for finish their execution.

Next we will deal with two different aspects for the exposed general idea: the phases synchronization and finally efficiency analysis.

Synchronization Design

Accordingly whit the previous explanation, these phases shared between the two different processes types, evolution rules and membrane, as it can be observed in tables 1 and 2.

```
(1)   Phase 1: Membrane initialization
(2)   REPEAT
(3)     Phase 5: Proposal Management &
                 Last active rule maximal application
(4)     Phase 7: Checking membrane halt
(5)   UNTIL End
```

Table 1: Process Type Membrane

```
(1)   Phase 2: Rules initialization
(2)   REPEAT
(3)     Phase 3: Multiset proposition
(4)     Phase 4: Sum of Multiset Proposals
(5)     Phase 6: Checking rules halt
(6)   UNTIL End
```

Table 2: Process Type Evolution Rule

Both processes types are not disjoint and they must preserve the following synchronizations (Fig. 1 presents the activity diagram showing the needed synchronization in the different phases for the process membrane and two evolution rules processes):

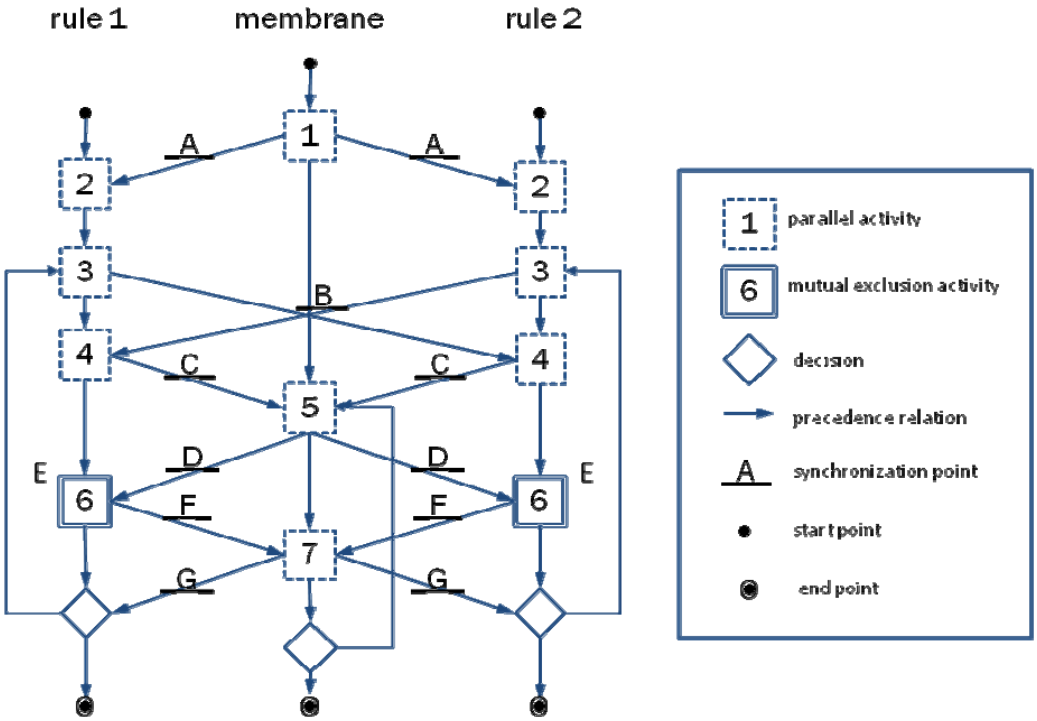


Figure 1: The activity diagram showing the needed synchronization in the different phases for the membrane and two evolution rules

A. Every evolution rule must wait for initialization until membrane initialization finishes.

B. Each evolution rule must wait for their neighbor evolution rules finish their respective additions of proposed multisets by their neighbor evolution rules.

C. Membrane must wait to start management collision until evolution rules finish accumulating the proposed multisets.

D. Every evolution rule must wait to start checking halting condition until membrane finishes multisets subtraction.

E. Every evolution rule must wait for the mutual exclusion to access into the active evolution rule data structure and it can perform its register for the next loop iteration.

F. Membrane must wait to checking halting condition until evolution rules finish their corresponding checking for halting conditions.

G. Every evolution rule must wait to start to determine, it they finish their execution or come back to propose a new multiset, until membrane halt checking finishes.

Efficiency Analysis

Analyzing the membrane process it can be observed that the proposed algorithm executes, at the most, so many times as rules exist initially (we will denominate it by R), since in each iteration at least one rule is eliminated in the worse case. The rest of operations executed by the process membrane can be considered like basic operations. Moreover, the operations executed by the rules processes are simple, except the sum of proposals made in phase 4. This sum is performed two by two by neighborhood, reason why the obtained complexity order is $log_2 R_i$, being R_i the number of active rules minus one in each loop iteration. Consequently, the complexity order of the proposed algorithm is:

$$\sum_{i=R-1}^{2} log_2 i = log_2(R - 1) + log_2(R - 2) + \ldots + log_2 2 = log_2(R - 1)!$$

Consequently, we can conclude that the complexity order of the proposed algorithm -in the worse case- is $log_2(R - 1)$!, but better results can be expected experimentally than the ones obtained theoretically, because exists the possibility that in a same loop iteration disappears more than a rule.

Future Work

In first phase of the presented algorithm -membrane initialization- a global probability for the rule processes is determined. This value determines the probability of proposing an objects multiset by a rule. Assigning a value of $1/R$ to this probability very good results in the made tests have been obtained. At the moment we are working in the process of determination of this value, trying of obtaining a better efficiency.

In addition, since one has studied in this work, the presented algorithm eliminates an active rule in each loop iteration. Evidently, the order in that the rules are applied influences in the final results obtained. Therefore, to improve the algorithm efficiency it seems interesting to study as it would have to be the last applied rule, studying the relations between the antecedents of the rules available.

Conclusions

This paper introduces an algorithm of active rules application based on rules elimination in transition P systems. The two most important characteristics of this algorithm are:

- The presented algorithm is massively parallel
- The execution time of the algorithm is time delimited, because it only depends on the number of rules of the membrane. The number of rules of the membrane is a well-known static information studying the P system

We think that the presented algorithm can represent an important contribution in particular for the problem of the application of rules in membranes, because it presents high productivity and it allows estimate the necessary time to execute an evolution step. Additionally, this last one allows to make important decisions related to the implementation of P systems, like the related ones to the software architecture.

Bibliography

[Ciobanu, 2002] G. Ciobanu, D. Paraschiv, "Membrane Software. A P System Simulator". Pre-Proceedings of Workshop on Membrane Computing, Curtea de Arges, Romania, August 2001, Technical Report 17/01 of Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain, 2001, 45-50 and Fundamenta Informaticae, vol 49, 1-3, 61-66, 2002.

[Ciobanu, 2004] G. Ciobanu, G. Wenyuan, "A P system running on a cluster of computers", Proceedings of Membrane Computing. International Workshop, Tarragona (Spain). Lecture Notes in Computer Science, vol 2933, 123-150, Springer Verlag, 2004.

[Ciobanu, 2006] G. Ciobanu, M. Pérez-Jiménez, Gh. Păun, "Applications of Membrane Computing". Natural Computing Series, Springer Verlag, October 2006.

[Fernández 2006] L. Fernández, F. Arroyo, J. Tejedor, J. Castellanos. "Massively Parallel Algorithm for Evolution Rules Application in Transition P System". Seventh Workshop on Membrane Computing, WMC7, Leiden (The Netherlands). July, 2006

[Păun, 1998] G. Păun. "Computing with Membranes". In: Journal of Computer and System Sciences, 61(2000), and Turku Center of Computer Science-TUCS Report nº 208, 1998.

[Păun, 2005] G. Păun. "Membrane computing. Basic ideas, results, applications". In: Pre-Proceedings of First International Workshop on Theory and Application of P Systems, Timisoara (Romania), pp. 1-8, September, 2005.

[Tejedor, 2006] J. Tejedor, L. Fernández, F. Arroyo, G. Bravo. "An Architecture for Attacking the Bottleneck Communications in P systems". In: Artificial Life and Robotics (AROB 07). Beppu (Japan), January 2007.

[Tejedor, 2007] J. Tejedor, L. Fernández, F. Arroyo, A. Gutiérrez. "Algorithm of Active Rules Elimination for Evolution Rules Application" (submited). In 8th WSEAS Int. Conf. on Automation and Information, Vancouver (Canada), June 2007.

Authors' Information

F. Javier Gil Rubio - Dpto. de Organización y Estructura de la Información, E.U. de Informática. Natural Computing Group, Universidad Politécnica de Madrid, Spain; e-mail: jgil@eui.upm.es

Luis Fernández Muñoz - Dpto. de Lenguajes, Proyectos y Sistemas Informáticos, E.U. de Informática. Natural Computing Group, Universidad Politécnica de Madrid, Spain; e-mail: setillo@eui.upm.es

Fernando Arroyo Montoro - Dpto. de Lenguajes, Proyectos y Sistemas Informáticos, E.U. de Informática. Natural Computing Group, Universidad Politécnica de Madrid, Spain; e-mail: farroyo@eui.upm.es

Jorge A. Tejedor Cerbel - Dpto. de Organización y Estructura de la Información, E.U. de Informática. Natural Computing Group, Universidad Politécnica de Madrid, Spain; e-mail: jtejedor@eui.upm.es

RESEARCHING FRAMEWORK FOR SIMULATING/IMPLEMENTATING P SYSTEMS

Sandra Gómez, Luis Fernández, Iván García, Fernando Arroyo

Abstract: Researching simulation/implementation of membranes systems is very recent. Present literature gathers new publications frequently about software/hardware, data structures and algorithms for implementing P system evolution.

In this context, this work presents a framework which goal is to make tasks of researchers of this field easier. Hence, it establishes the set of cooperating classes that form a reusable and flexible design for the customizable evaluation with new data structures and algorithms. Moreover, it includes customizable services for correcting, monitoring and logging the evolution and edition, recovering, automatic generating, persistence and visualizing P systems.

Keywords: P System, framework, simulation, implementation.

ACM Classification Keywords: D.1.m Miscellaneous – Natural Computing

Introduction

P systems are a new computational model based on the membrane structure of living cells. This model has become, during last years, a powerful framework for developing new ideas in theoretical computation. "P systems with simple ingredients (number of membranes, forms and sizes of rules, controls of using the rules) are Turing complete" [Păun, 1999]. Moreover, P systems are a class of distributed, massively parallel and non-deterministic systems.

"As there do not exist, up to now, implementations in laboratories (neither in vitro or in vivo nor in any electronical medium), it seems natural to look for software tools that can be used as assistants that are able to simulate computations of P systems" [Ciobanu, 2006]. "An overview of membrane computing software can be found in literature, or tentative for hardware implementations, or even in local networks is enough to understand how difficult is to implement membrane systems on digital devices" [Păun, 2005]. Moreover, he says: "we avoid to plainly say that we have 'implementations' of P systems, because of the inherent non-determinism and the massive parallelism of the basic model, features which cannot be implemented, at least in principle, on the usual electronic computer -but which can be implemented on a dedicated, reconfigurable, hardware [...] or on a local network" [Păun, 2005]. Thereby, there exists many simulators in bibliography but "the next generation of simulators may be oriented to solve (at least partially) the problems of storage of information and massive parallelism by using parallel language programming or by using multiprocessor computers" [Ciobanu, 2006].

The goal of this work is to present a framework to make easier the tasks of researchers who develop simulators/implementations of P systems. It does not expect to be a new simulator/implementation. It presents a set of cooperating classes that form a reusable design for developing simulators/implementations of P systems. This framework provides an architectonical guide to divide the design in abstract classes and to define their responsibilities and collaborations. Researchers have to adapt the framework to a concrete simulator/implementation inheriting and compounding instances of framework classes.

This paper is structured as follows: next section presents related works then, they are presented the requirements and design guidelines for the framework. Finally, conclusions are presented.

Related Works

Membrane system implementation is a very recent investigation field. First approaches were simulators [Ciobanu, 2006] that demonstrated the functionality of the membrane systems. But, they lacked distributed and massively character.

First distributed implementations are presented in [Syropoulos, 2003] and [Ciobanu, 2004]. In their distributed implementations of P systems use Java Remote Method Invocation (RMI) and the Message Passing Interface (MPI) respectively, on a cluster of PC connected by Ethernet. These last authors do not carry out a detailed analysis of the importance of the time used during communication phase in the total time of P system evolution, although Ciobanu affirms that "the response time of the program has been acceptable. There are however executions that could take a rather long time due to unexpected network congestion" [Ciobanu, 2003]. In [Tejedor, 2007a] [Bravo, 2007a] [Bravo, 2007b], it is determined that the problem in implementing P systems is the time necessary in the communication of multisets among membranes allocated in different devices (PCs, PICs, or chips). This fact, forces to resign parallelism to the maximum to as much reach a parallelism degree dependent of the speed of the communications and the application of the evolution rules. Therefore, it is necessary to develop faster application algorithms that adapt so much to the sequential technologies as to the parallel ones.

On the other hand, [Fernández, 2007] determines the appropriate software architecture that is executed over a given evolution P System hardware architecture. So, it pretends to determine the set of process and their relationships that are appropriate to be executed over a set of connected processors. Considered possibilities are: evolution rules oriented software architecture, membranes oriented software architecture and processors oriented software architecture.

Works of investigation about sequential and/or parallel algorithms designed for the different phases of the evolution of a P system are very varied: for the utility of the evolution rules: [Frutos, 2007]; for the applicability of evolution rules: [Fernández, 2006a]; for the application of evolution rules: [Fernández, 2006b] [Fernández, 2006c] [Tejedor, 2007b] [Tejedor, 2007c] [Gil, 2007].

With respect to the storage of the information of a P System, [Fernández, 2005a] defines a universal vocabulary with XML technology and [Gutiérrez, 2007b] presents new data structures that compress multisets of objects information without penalizing the basic operations on these.

Finally, it is possible to indicate the works on different technologies whose objective is to implant the different architectures, algorithms and previous data structures. Thus, we found a line about circuits hardware in [Petreska, 2003] [Arroyo, 2004a] [Arroyo, 2004b] [Arroyo, 2004c] [Fernández, 2005b] [Martínez, 2006a] [Martínez, 2006b], the new opened line about microcontrollers in [Gutiérrez, 2006] [Gutiérrez, 2007a], and the traditional line about personal computers in [Syropoulos, 2003] [Ciobanu, 2004].

Requirements

In this context, pursued goals is to develop a highly reusable framework that is flexible enough for any researcher to be able of concentrating on developing the algorithm or data structure object of its investigation.

In this line, the framework provides to the researchers the following reusable modules:

1.1 Implementation of every standard data structure, agreed to the specification model, in order to equip framework with the total functionality of a simulator of membrane system. Hence, it is had the data structures for the symbols, multisets, evolution rules and membranes.

1.2 Implementation of every standard algorithm, agreed to the specification model, in order to equip framework with the total functionality of a simulator of membrane system. Hence, it is had the algorithms for utility phases, applicability, activity, application, communication and dissolution.

1.3 Process management and synchronization for the different software architectures: evolution rules oriented, membranes oriented and processor oriented.

1.4 Management of automatic detection of errors of any algorithm for functional tests.

1.5 Management of automatic monitoring (time, space, number of operations, …) of any algorithm for non functional tests.

1.6 Management of persistence, visualizing and logs for tracking the algorithms.

1.7 Management of P System for its edition, recovery, automatic generation of parameterized sets of tests

1.8 Management of configurations for P systems evolution.

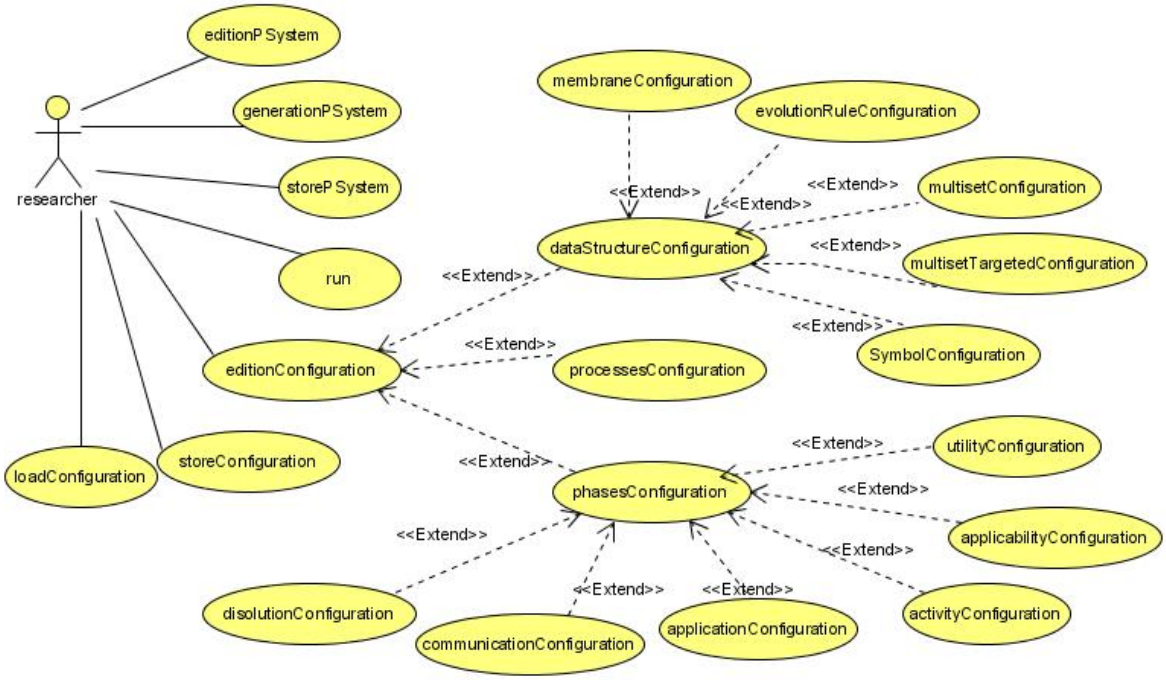


Figure 1: Use Case Diagram.

On the other hand, the framework provides to the researchers the following flexibility for a given evolution of the P System:

2.1. Extension by inheritance of new data structures for the symbols, multisets, evolution rules and membranes.

2.2. Extension by inheritance of new algorithms for utility, applicability, activity, application, communication and dissolution phases.

2.3. Extension by inheritance of new functionalities over P systems (analyisis, compilation, …).

2.4. Architecture process, phases, algorithms and data structures configuration.

2.5. Evolution, visualization, monitorization, correction and logs configuration.

Figure 1 shows the use case diagram corresponding to the previous requirements.

Framework

Figure 2 shows a class diagram of the domain model that has the most important object classes according to P System specification.

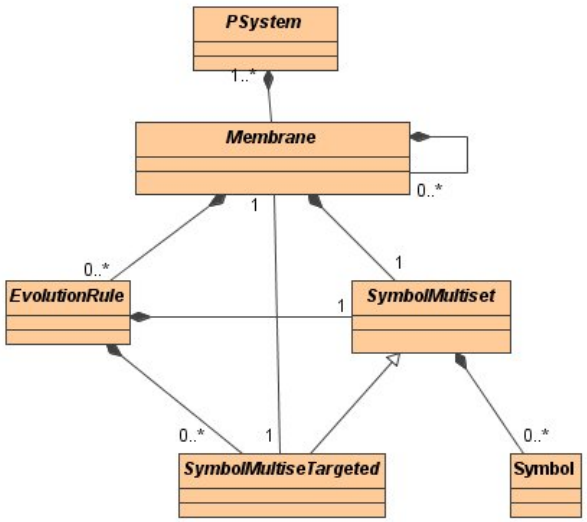


Figure 2: P system Domain Model Class Diagram.

Figure 3 shows the class diagram that was designed for covering requirements 1.1 and 2.1. This way, concrete classes in the third level of the inheritance hierarchy are contributed for every standard data structure. Also, it makes easy the incorporation of new data structures inheriting from the abstract classes of the second inheritance hierarchy level.

In particular, class *ElementFactory* is responsible of the configuration of the data structures for a given evolution of requirement 2.4.

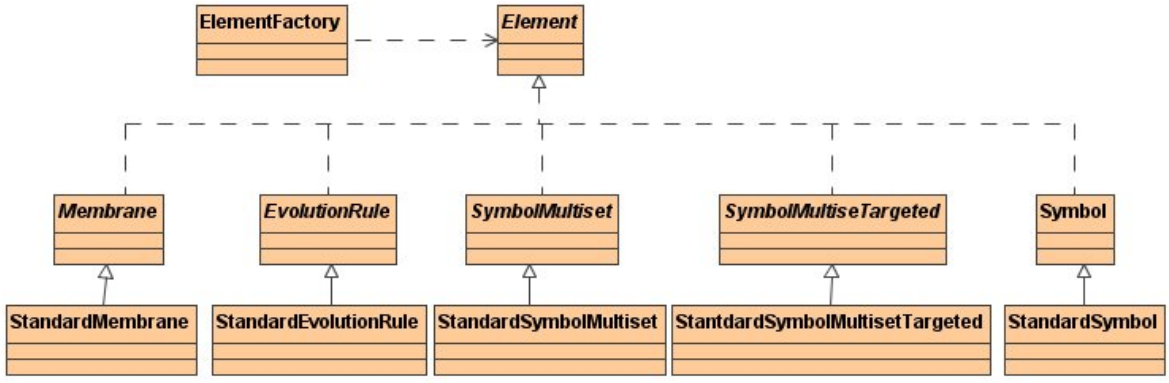


Figure 3: Data Structures Class Diagram.

Figure 4 shows the class diagram designed to cover requirements 1.2 and 2.2. This way, concrete classes in the fifth level of the inheritance hierarchy are contributed for every algorithm of evolution phases. Also, it makes easy the incorporation of new algorithms inheriting from forth level of inheritance hierarchy abstract classes.

In particular, class PhaseFactory is responsible of the configuration of the phases and algorithms for a concrete evolution of requirement 2.4.

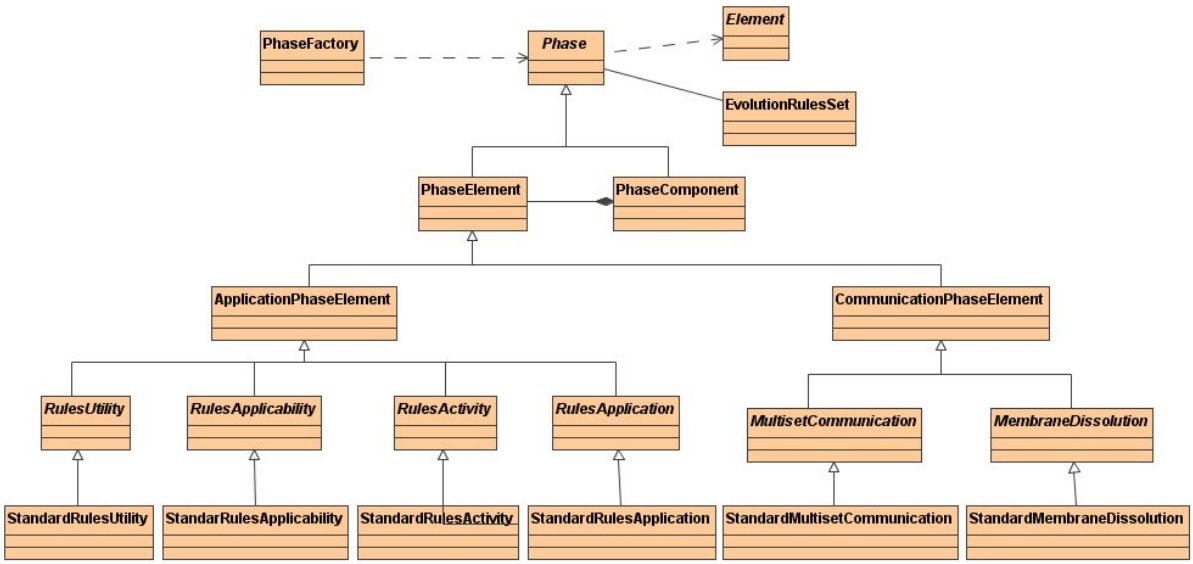


Figure 4: Algorithms Class Diagram.

Figure 5 shows the class diagram designed to cover the requirement 1.3. This way, concrete classes in the second level of the inheritance hierarchy are contributed for every process architectures together with the classes for the process synchronization.

In particular, class *ProcessFactory* is responsible of the configuration of the process architectures for a concrete evolution of requirement 2.4.

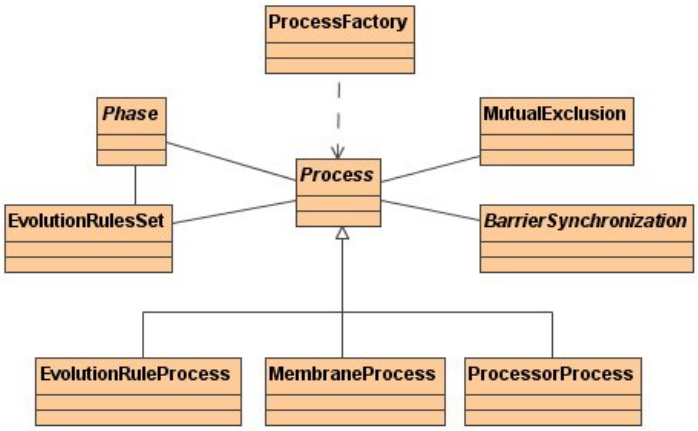


Figure 5: Software Architecture Class Diagram.

Figure 6 shows the class diagram designed to cover requirement 1.4, 1.5, 1.6 and 2.3. This way, concrete classes in the forth level of inheritance hierarchy are contributed for the detection of errors and automatic monitorization of functional and non functional sets of tests respectively, and for the persistence, log and visualization of the results of a given evolution. Moreover, new functionalities (P System analysis, compilation, …) can be developed inheriting from *VisitorElement*.

In particular, class *VisitorFactory* is responsible of the configuration of a given evolution of the requirement 2.5.

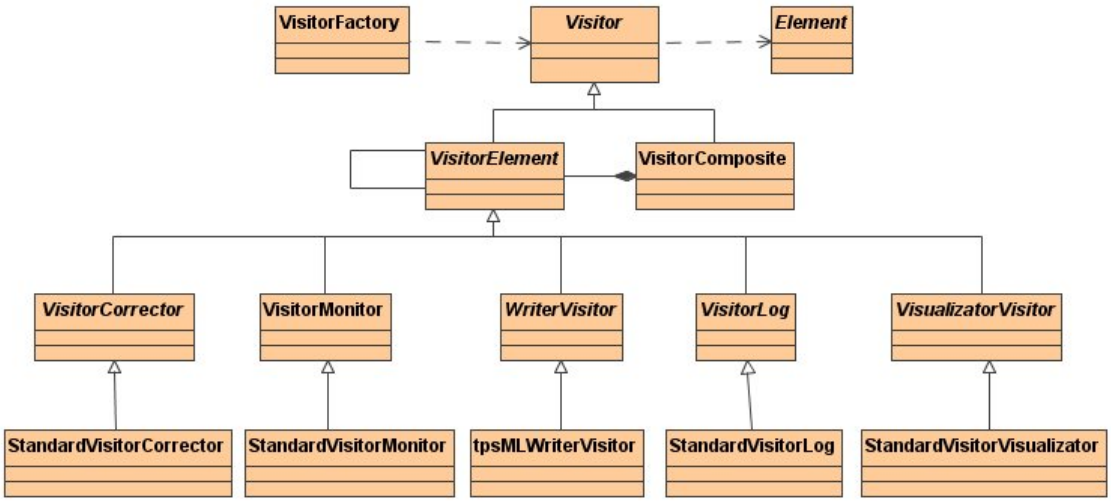


Figure 6: Visitors Class Diagram

Figure 7 shows the general class diagram that relates every class of previous class diagrams. In particular, class *PSystemFactory* is responsible of the edition, recovery and sets of parameterized tests automatic generation of requirement 1.7. Moreover, set of factory classes is responsible of managing the configurations for a given evolution of requirement 1.8.

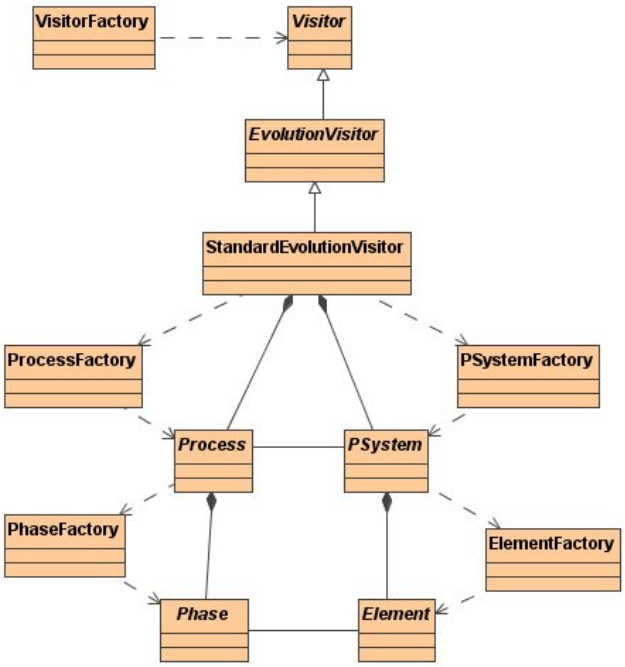


Figure 7: General Class Diagram.

Conclusion

This work contributes a framework that makes investigation of developing new simulators and implementations of membrane systems easier. Its goal is to provide enough reusability and flexibility to get the researcher is concentrated in the goals of his investigation. This way, it is possible to reuse standard data structures and algorithms of the P System model, processes and synchronization management, error detection, monitorization and log for tests phase and the edition, recovery, automatic generation, persistence and visualization of P systems. On the other hand, the simple inheritance mechanism provides flexibility for the incorporation of new data structures and algorithms.

Bibliography

[Alonso, 2007] S. Alonso, L. Fernández, F. Arroyo, J. Gil. A Circuit Implementing Massive Parallelism in Transition P Systems. Fifth International Conference Information Research and Applications (i.TECH-2007). Varna (Bulgary) june, 2007. (submitted).

[Arroyo, 2004a] F. Arroyo, C. Luengo, Castellanos, L.F. de Mingo. A binary data structure for membrane processors: Connectivity Arrays. Lecture Notes in Computer Science (A. Alhazov, C. Martin-Vide, G. Mauri, G. Paun, G.Rozenberg, A. Saloma, eds.) Springer Verlag (2933), 2004. 19-30.

[Arroyo, 2004b] F. Arroyo, C. Luengo, Castellanos, L.F. de Mingo. Representing Multisets and Evolution Rules in Membrane Processors. Preproceedings of the Fifth Workshop on Membrana Computing (WMC5). Milano (Italy) june, 2004. 126-137.

[Arroyo, 2004c] F. Arroyo, C. Luengo, L. Fernandez, L.F. de Mingo, J. Castellanos, Simulating membrane systems in digital computers. ITHEA-2004 International Journal Information Theories and Applications (vol.11 - 1) 2004. 29-34.

[Bravo, 2007a] G. Bravo, L. Fernández, F. Arroyo, J.A. Frutos. A Hierarchical Architecture with parallel comunication for implementing P Systems. Fifth International Conference Information Research and Applications (i.TECH-2007). Varna (Bulgary) june, 2007. (submitted).

[Bravo, 2007b] G. Bravo, L. Fernández, F. Arroyo, J. Tejedor. Master/Slave Parallel Architecture for Implementing P Systems. The 8th WSEAS International Conference on Mathematics and Computers in Business and Economics (MCBE'07). Vancouver (Canada) june, 2007. (submitted).

[Ciobanu, 2004] G.Ciobanu, W.Guo . P Systems Running on a Cluster of Computers. Workshop on Membrane Computing (Gh. Păun, G. Rozenberg, A. Salomaa Eds.), LNCS 2933, Springer, 123-139.

[Ciobanu, 2006] G. Ciobanu, M. Pérez-Jiménez, Gh. Păun. Applications of Membrana Computing". Natural Computing Series, Springer Verlag, october, 2006.

[Fernández, 2005a] L. Fernández, F. Arroyo, J. Castellanos, V.J. Martínez, L.F. Mingo. Software Tools/ P System Simulators Interoperability. (R. Freund, G. Lojka, M. Oswald, Gh. Paun, eds.) Preproceedings of Sixth International Workshop on Membrane Computing (WMC6), Vienna, (Austria) june, 2005. 147-161.

[Fernández, 2005b] L. Fernandez, V.J. Martínez, F. Arroyo, L.F. Mingo. A Hardware Circuit for Selecting Active Rules in Transition P Systems. (G. Ciobanu, Gh. Paun, eds.) Preproceedings of First International Workshop on Theory and Application of P Systems, Timisoara (Romania), september, 2005. 45-48.

[Fernández, 2006a] L. Fernández, F. Arroyo, I. García, G. Bravo. Decision Trees for Applicability of Evolution Rules in Transition P System. ITHEA-2006 Interantional Journal Information Theories and Applications (vol.11 - 1) 2006. 29-34.

[Fernández, 2006b] L. Fernández, F. Arroyo, J.A. Tejedor, J. Castellanos, Massively Parallel Algorithm for Evolution Rules Application in Transition P Systems. Preproceedings of Membrane Computing, International Workshop (WMC7), Leiden (The Netherlands) july, 2006. 337-343.

[Fernández, 2006c] L. Fernández, F. Arroyo, J. Castellanos, J.A. Tejedor, I. García, New Algorithms for Application of Evolution Rules based on Applicability Benchmarks. International Conference on Bioinformatics and Computational Biology(BIOCOMP06), Las Vegas (EEUU), july, 2006.

[Fernández, 2007] L. Fernández, F. Arroyo, I. García, A. Gutiérrez. Parallel software architectures analysis for implementing P systems. (M. Sugisaka, H. Tanaka, eds.), Proceedings of the 12th Int. Symposium on Artificial Life and Robotics (AROB07), Beppu (Japan) january, 2007. 494-499.

[Frutos, 2007] J.A.Frutos, L. Fernández, F.Arroyo, G.Bravo. Static Analysis of Usefulness States in Transition P Systems. Fifth International Conference Information Research and Applications (i.TECH-2007). Varna (Bulgary) june, 2007.

[Gil, 2007] F.J. Gil, L. Fernández, F. Arroyo, J.A. Tejedor. Delimited Massively Parallel Algorithm based on Rules Elimination for Application of Active Rules in Transition P Systems. Fifth International Conference Information Research and Applications (i.TECH-2007). Varna (Bulgary) june, 2007.

[Gutiérrez, 2006] A. Gutiérrez, L. Fernández, F. Arroyo, V. Martínez. Design of a Hardware Architecture based on Microcontrollers for the Implementation of Membrane Systems. Proc. on 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC-2006). Timisoara (Romania) septiembre, 2006. 39-42.

[Gutiérrez, 2007a] A. Gutiérrez, L. Fernández, F. Arroyo, S. Alonso. Hardware and Software Architecture for Implementing Membrane Systems: A case of study to Transition P Systems. The International Meeting on DNA Computing (DNA13), Memphis (USA) June 3-8, 2007. (submitted)..

[Gutiérrez, 2007b] A. Gutiérrez, L. Fernández, F. Arroyo, G. Bravo. Compression of Multisets and Evolution Rules Optimizing the Storage and Communication in Membrana System. Eight Workshop on Membrane Computing (WMC8). Thessaloniki (Greece) june, 2007 (submitted).

[Martínez, 2006a] V. Martínez, L. Fernández, F. Arroyo, I. García, A. Gutiérrez. *A HW circuit for the application of Active Rules in a Transition P System Region*. Proceedings on Fourth International Conference Information Research and Applications (i.TECH-2006). Varna (Bulgary) June, 2006. pp. 147-154. ISBN-10: 954-16-0036-0.

[Martínez, 2006b] V. Martínez, L. Fernández, F. Arroyo, A. Gutiérrez. *HW Implementation of a Bounded Algorithm for Application of Rules in a Transition P-System*. Proceedings on 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC-2006). Timisoara (Romania) septiembre, 2006. pp. 32-38.

[Păun, 1999] Gh. Păun. *Computing with membranes*. An introduction. Bulletin of the EATCS, 67, 139-152, 1999.

[Păun, 2005] Gh. Păun. *Membrane computing. Basic ideas, results, applications.* Pre-Proceedings of First International Workshop on Theory and Application of P Systems. Timisoara (Romania), september , 2005. 1-8.

[Petreska, 2003] B. Petreska and C. Teuscher. *A hardware membrane system*. A. Alhazov, C. Martin-Vide, Gh. Paun (eds.):Pre-proceedings of the workshop on Membrane Computing Tarragona, July 17-22 2003, 343-355.

[Syropoulos, 2003] A. Syropoulos, E.G. Mamatas, P.C. Allilomes. *A distributed simulation of P systems.*(A. Alhazov, C. Martin-Vide and Gh. Păun, eds.) Preproceedings of the Workshop on Membrana Computing. Tarragona (Spain), july, 2003, 455-460.

[Tejedor, 2007a] A. Tejedor, L. Fernández, F. Arroyo, G. Bravo, *An architecture for attacking the bottleneck communication in P systems*. In: M. Sugisaka, H. Tanaka (eds.), Proceedings of the 12th Int. Symposium on Artificial Life and Robotics, Jan 25-27, 2007, Beppu, Oita, Japan, 500-505.

[Tejedor, 2007b] A. Tejedor, L. Fernández, F. Arroyo, A. Gutiérrez. *Algorithm of Active Rule Elimination for Application of Evolution Rules.* The 8th WSEAS International Conference on Mathematics and Computers in Business and Economics (MCBE'07). Vancouver (Canada) june, 2007. (submitted).

[Tejedor, 2007c] A. Tejedor, L. Fernández, F. Arroyo, S. Gómez. *Application Algorithm based on Evolution Rules Competitivity.* Eight Workshop on Membrane Computing (WMC8). Thessaloniki (Greece) june, 2007 (submitted).

Authors' Information

Sandra María Gómez Canaval – Natural Computing Group. Dpto. Organización y Estructura de la Información de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: sgomez@eui.upm.es

Luis Fernández Muñoz – Natural Computing Group. Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: setillo@eui.upm.es

Iván García – Natural Computing Group. Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: igarcia@eui.upm.es

Fernando Arroyo – Natural Computing Group. Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: farroyo@eui.upm.es

GRID INFRASTRUCTURE FOR SATELLITE DATA PROCESSING IN UKRAINE

Nataliia Kussul, Andrii Shelestov, Mykhailo Korbakov, Oleksii Kravchenko, Serhiy Skakun, Mykola Ilin, Alina Rudakova, Volodymyr Pasechnik

Abstract. In this paper conceptual foundations for the development of Grid systems that aimed for satellite data processing are discussed. The state of the art of development of such Grid systems is analyzed, and a model of Grid system for satellite data processing is proposed. An experience obtained within the development of the Grid system for satellite data processing in the Space Research Institute of NASU-NSAU is discussed.

Keywords: Grid system, satellite data processing, Grid services.

ACM Classification Keywords: H.3.4 Systems and Software - Distributed systems, H.3.3 Information Search and Retrieval.

1 Introduction

Grid systems, originated by Ian Foster [[1], are becoming standard solutions for enabling remote computations execution and distributed data access and processing in environments of the different level of scalability.

The aim of Grid system could be formulated as connecting data, processing powers and algorithms that distributed over the network for solving particular problems. Grid system should be universal up to some degree, so these problems should not be hardcoded during its development. Instead, a set of problems being solved in a Grid environment must be open for modifications and addition of new ones. This goal is achieved by introducing standard interfaces for communicating between different kinds of Grid resources and clients.

Space agencies all over the world are successfully working on development of Grid technology for their application areas. This is due to the fact that Earth observation (EO) domain is characterized by the acquisition of large amounts of data from satellites and distributed nature of data. Furthermore, the single EO product and the data after its initial processing may easily exceed the gigabyte size. Thus, problems of storing, indexing for quick retrieval on application's demand as well as distributed computing arise within the above mentioned area. Grid technology can provide comprehensive solutions for this problem.

In this paper a brief overview of Grid systems for satellite data processing is given. Common approaches and conceptual foundations of development of Earth Observation Grid system are defined. A model of Grid system for satellite data processing is proposed and verified based on a test-bed of Grid system for satellite data processing that was developed in the Space Research Institute of the National Academy of Sciences of Ukraine and the National Space Agency of Ukraine.

2 Overview of Grid Systems for Satellite Data Processing

Nowadays Grid technology is widely applied for the solution of various problems in many domains [[2]. These applications span a wide spectrum. In this section we give a brief overview of Grid systems that are used for satellite data processing.

Earth Science GRID on Demand project [http://eogrid.esrin.esa.int/] is being developed by European Space Agency (ESA) and European Space Research Institute (ESRIN). GRID is considered as a comfortable "open platform" for handling computing resources, data, tools, etc., and not limited to only high performing computing. Online access to different data is enabled within this project, in particular to data provided by various instruments of Envisat satellite [http://envisat.esa.int/], the SEVIRI instrument onboard MSG (the Meteosat Second Generation) satellite, ozone profiles derived from GOME instrument, etc. One of the most important applications is the analysis of long-term data. For example, the analysis of 8 years of GOME on-board temperatures (overall 525 Gb of data) took less than 2 days on 40 computer elements of ESRIN "Grid-on-demand" structure (overall 38460 files were processed).

Grid Web Portal provides access to the "Grid-on-demand" [http://eogrid.esrin.esa.int/] resources enabling:

— Personal certification

— Time /space selection of data, directly from the ESA catalogue

— Data transfer from ESA data storages

— Job selection, launching and live status

— Visualization in OpenGIS Web Map and Google Earth

— Access to user products and documentation

Nowadays “Grid-on-demand” infrastructure consists of more than 150 working nodes with ability to store and handle of about 70 Gb of data. As middleware Globus Toolkit 2.4 and LCG/EGEE components are being used.

Japan Aerospace eXploration Agency (JAXA) [[3] and KEIO University started to establish “Digital Asia” system aimed at semi-real time data processing and analyzing. They use GRID environment to accumulate knowledge and know-how to process remote sensing data. The problems of radiometric rectification and composition of remotely sensed data are being solved.

National Aeronautics and Space Administration (NASA) have created Information Power Grid (IPG) [[4] targeting an operational Grid environment incorporating major computing and data resources at multiple NASA sites in order to provide an infrastructure capable of routinely addressing larger scale, more diverse, and more transient problems than is possible today. One of the problems being solved is development of techniques for satellite data fusion. Nowadays IPG have approximately 600 CPU nodes of Computing resources and 30-100 Terabytes of archival information/data storage resources.

Spatial Information Grid (SIG), a research project supported by 863 projects of China government, is a series of special grid researches in the filed of Earth Observation. SIG has been designed to be the tested of grid middleware research and grid-enable spatial information services and applications. There are 12 data centers have been involved SIG. The Web Portal has been developed in order to provide access to SIG resources (http://159.226.224.52:6140/Grid/application/index_en.jsp). This portal enables geo-data discover and processing, work monitoring, and grid resources (all service/job/node etc.) management.

3 Why EO Domain requires Grid

In particular, EO domain is characterized by the acquisition of large amounts of data from satellites. For example, an image acquired from ETM+ instrument from the Landsat-7 satellite is approximately 700 megabytes in size. NASA is planning to launch National Polar-orbiting Operational Environmental Satellite System (NPOESS) project [[5] that in 5 years will generate approximately 1 petabytes of information.

In general EO domain is characterized by:

— large amounts of data acquired from different satellites in different spectral bands that need to be integrated with aerial and in-situ components and maps;

— thematic problems solving require the use of data from multiple sources which in turn leads to the need of use complex data fusion and data mining techniques;

— long-term archives need to be created with uniform access to them.

To enable processing and management of such volumes of data sets and information flows an appropriate infrastructure is needed that will support the following functionality:

— access to distributed resources (data/services/network/computing/storage);

— high flexibility, to foster data fusion and assimilation (meteo, models, global changes, etc.);

— portal enabling easy and homogeneous accessibility;

— virtual organisation (VO) Management;

— collaborative work (e.g. sharing of data sources, tools, means, models, algorithms);

— seamless integration of resources and processes;

— allow processing of large historical archives;

— avoid unauthorised access to/use of resources.

Grid technology is an appropriate solution for solving such kind of problems.

4 The Architecture of Systems for Satellite Data Processing

Based on the existing systems and the systems that are currently being developed, it is possible to identify principal components (sub-systems) and informational flows within a system for satellite data processing (Fig. 1).

Data Storage Sub-System is intended for gathering data from multiple sources, i.e. aerial and space-borne data, in-situ data, etc. Usually, the storage system is organized as multi-layer system each level being characterized by different frequencies of data use. We will consider three-level architecture that consists of an operational archive, a short-term data archive, and a long-time archive. The operational archive contains information that was obtained recently, and there is a higher possibility of accessing this kind of data by users. To store such data hard-discs are usually used enabling minimum access time to data. The short-term archive contains data that were obtained weeks or some months ago. To store these data tape-drives are used. The long-term archive contains data obtained years ago. In some cases such kind of archives can be not automated. They can also use high level of data compression and slow recorders. Time access to these archives can be of hours or days. Such three-level architecture of data storage system is implemented in archives of NASA (USA), DLR (Germany), JAXA (Japan). Two-level architecture is used in the State Research & Productive Center "Pryroda" (Ukraine).

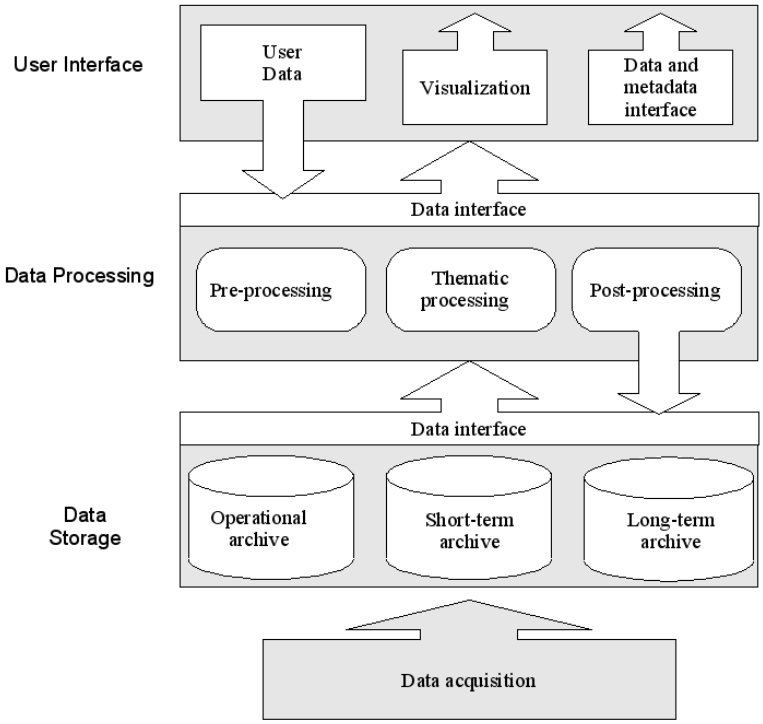


Fig. 1. Three-level architecture of system for satellite data processing

Data Processing Sub-System is intended for data pre-processing (e.g. radiometric and geometric correction of space images, filtering, etc.) and thematic problems solving based on different models and data integration from multiple sources.

User Interface Sub-System is a front-end component that allows end-users to interact with the system. This system is intended for delivering products and services (e.g. raw data and different levels of processed data delivery) to end-users on regular basis or based on their request.

5 Grid Infrastructure for Satellite Data Processing in Space Research Institute of NASU-NSAU

5.1 Grid Infrastructure

A Grid system for satellite data processing that integrates resources of the Space Research Institute of NASU-NSAU, the Institute of Cybernetics of NASU, and the State Research & Productive Center "Pryroda" has been developed. The Grid system consists of two computational SCIT-clusters (the Institute of Cybernetics), a cluster of the Space Research Institute, and an archive of the Meteosat satellite images acquiring from the data center "Pryroda". The developed infrastructure also includes works-stations and network data storage elements. Figure 2 illustrates the overall system architecture.

The developed Grid system provides both informational and computational resources of Space Research Institute and Institute of Cybernetics. The computational resources compromise SCIT-1 (48 processors Intel Xeon) and SCIT-2 (64 processors Intel Itanium2) clusters belonging to the Institute of Cybernetics, and the cluster of the Space Research Institute that is used as testing environment. An interface between Grid system and the computational resources is enabled by Grid Resource Allocation and Management (GRAM) service of Globus

Toolkit 4. GRAM enables translation of RSL-XML format that is used for job submission request in Globus Toolkit 4 in a format of local job scheduling systems (PBS, Condor, LFS, etc.). Globus provides a set of adapters for standard local job scheduling systems, and tools enabling the development of new adapters. The cluster of thq Space Research Institute uses Torque job scheduling system that is PBS-compatible. In contrast, the SCIT-clusters use its own job scheduling system. That is why, a new adapter was developed in order to integrate these resources in the Grid system.

Up to this moment informational resources consist of archive where data acquired from the “Pryroda” centre and from Internet are stored. In the near future we are planning to provide access to the Meteosat Second Generation (MSG) satellite through DVB technology. The developed archive provides FTP and GridFTP interfaces. Currently, a multi-level data access OGSA-DAI interface is under development which will enable complex distributed requests execution and results combination.

A workflow in the Grid system consisting in job submissions, data transfers, proxy certificates renewal, etc., is controlled by scripts, written in Karajan language [[6]. Karajan is developed as workflow description language for Grid environments and possesses many useful features, such as transparent scheduling and job submission, declarative parallelism and easy extendibility by Java.

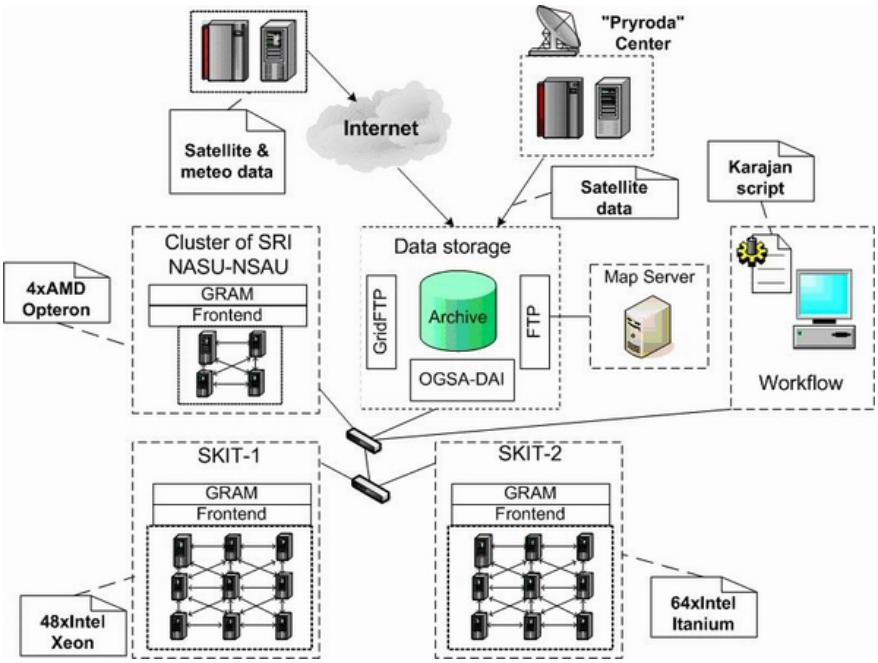


Fig. 2. Current Grid system infrastructure developed in the Space Research Institute of NASU-NSAU

5.2 Applied Services

The developed Grid system is currently used to process various satellite data, such as data acquired by the Meteosat satellite and the MODIS instrument onboard the Terra satellite.

Meteosat data in infrared spectrum are used in order to extract a cloud mask using Markov Random Field segmentation algorithm [[7] (Fig. 3). Image processing is done in three steps. First step consists in image filtration (namely, noise detection and removal) that is done using modified version of median filter [[8]. The second step is the segmentation of the image. The third step is post-processing and preparation of the data to be visualized by a map-server. The last step includes geo-reference of raw image and cloud mask, images re-projection, cloud boundary transformation in vector format, metadata creation for visualization. All these algorithms are implemented in the form of Web services available on http://www.dos.ikd.kiev.ua.

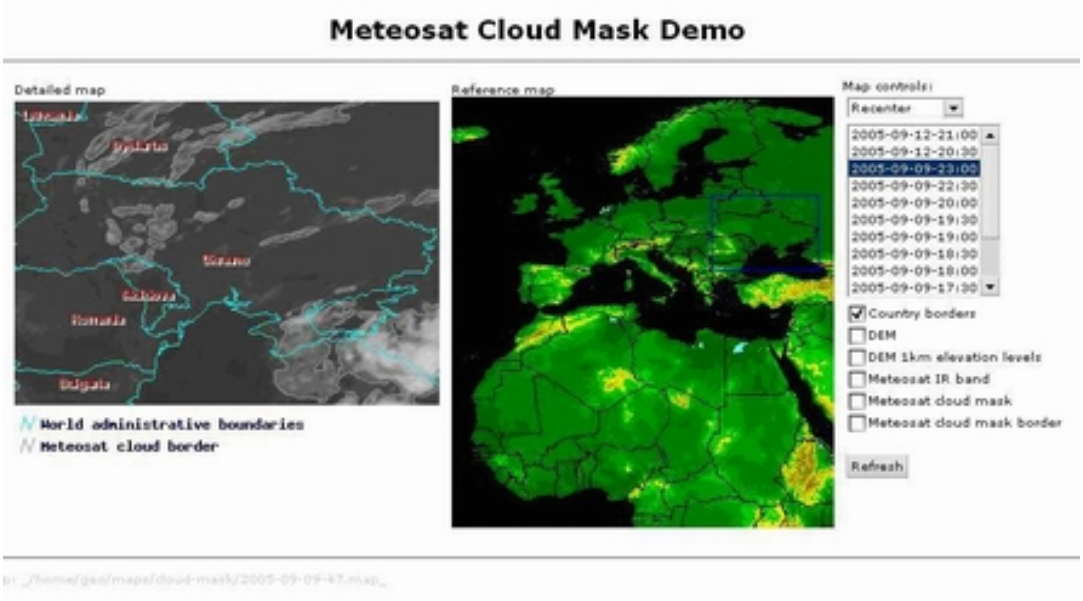


Fig. 3. Visualization of the Meteosat data processing

MODIS data are used for water quality monitoring in Dnieper river estuary. For this problem solving additional information is required such as in-situ measurements and a number of meteorological parameters, which are acquired using meteorological simulations. For this purpose we use WRF (Weather Research & Forecasting) mesoscale meteorological model [[9]. In order to provide initial and boundary conditions we use data produced by global meteorological model, namely Global Forecast System (GFS), and in-situ measurements. Currently we provide every 6 hours 3-day forecasts for the territory of Ukraine (Fig. 4).

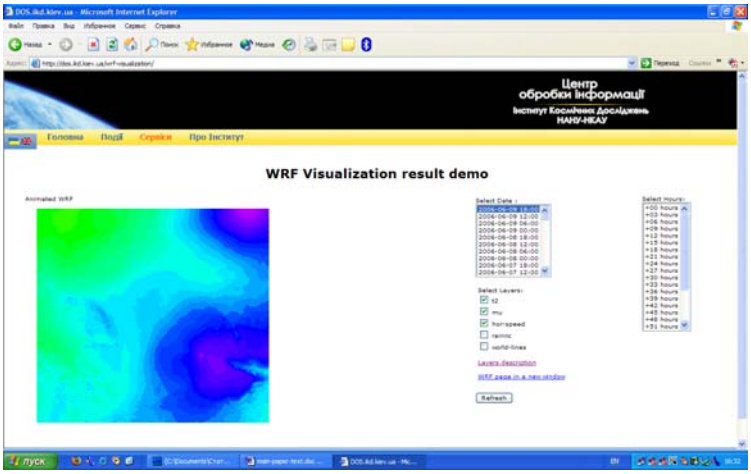


Fig. 4. WRF model visualization

The visualization of resulting data is done with the use of open-source UMN MapServer [[10] software that supports OGC (Open Geospatial Consortium) [[11] standards for spatial data representation.

6 Grid Infrastructure Simulation

Simulation is a common and useful approach for designing complex distributed systems with no exception to Grid. By using models one can decrease TCO (total cost ownership) and save funds on initial installation. Grid systems simulation requires appropriate software usage. The simulation of a Grid testbed in the Space Research Institute was performed by using GridSim [[12] modeling software. Different job scheduling algorithms were analyzed for independent tasks and for data-sharing tasks.

We used GridSim due to its ability to simulate common components of distributed systems such as heterogeneous resources, users, applications and Grid specific components including resource brokers and

schedulers for single and multiple administrative domains. Within GridSim package resources can be modeled using time- and space-sharing modes, thus representing workstations, SMP systems and clusters. There are other available Grid simulation packages, such as MicroGrid [[13] and SimGrid [[14]. However, GridSim is more flexible in model design, and does not impose additional requirements, such as Globus Toolkit installation.

Figure 5 illustrates GridSim class diagram where only redefined methods are depicted. New methods were added in order to extend basic GridSim functionality for our simulations. For example, Broker class was extended for Grid infrastructure resource brokering, GRIDTopology class for Grid infrastructure resource description and presentation.

GridSim model of the developed Grid system was used to estimate different job scheduling algorithms. Two common use cases were examined:

— a large group of independent tasks

— a set of tasks that are using common data

The first use case in comparison corresponds to the ideal parallel algorithm. All branches in algorithm can be executed independently in any order. This is a common situation in Monte-Carlo simulation or pixelwise image processing. The problems of scheduling for independent tasks are well investigated [[15]. However, these investigations stay in the field of homogeneous and static heterogeneous distributed computational systems. In turns, dynamic and heterogeneous Grid environments require some modifications to existing scheduling algorithms to take advantage of full utilization of system resources. The proposed algorithm is based on weighted factoring algorithm [[16]. The proposed modifications lie in using dynamic information about system's state to take into account side load on computational resources. Fig. 6a illustrates the performance of modified algorithm (bold line) comparing with traditional weighted factoring (thin line).

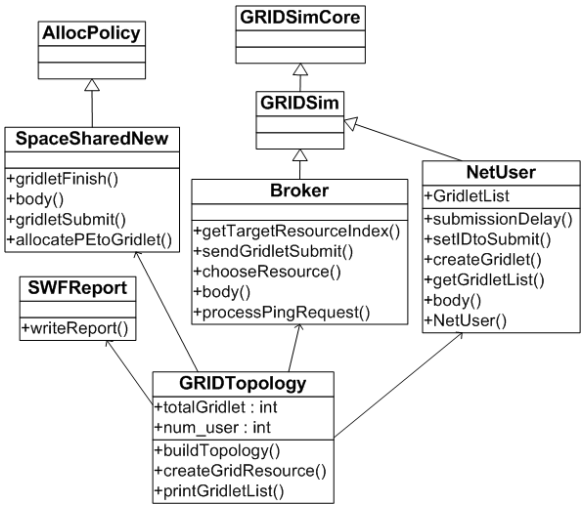


Fig. 5. GridSim Class Diagram

On each iteration of the modified algorithm a set of tasks from the group is assigned to some computational resource. The size of set is estimated as follows:

$$k_i(t) = \alpha \hat{\omega}_i(t) K(t) ,$$

where α is granularity parameter of algorithm, $\hat{\omega}_i(t)$ is the last-known load of computational resource, and $K(t)$ is a number of uncompleted tasks at a given time.

The last-known load $\hat{\omega}_i(t)$ is non-actual by its nature. There are always some lag between present moment and the moment when the information was last updated. The proposed algorithm is quite sensitive to these lags. The performance gain over unmodified version of the algorithm is lost when this parameter grows.

The second use case is a generalization of independent tasks case. The job now consists of tasks that need the same data of considerable size (transfer time of these data is comparable to total task execution time). The data granules are stored on the servers over the network. Each server has some limited bandwidth that separated between different transfers. The developed algorithm introduces fit measure U that shows a quality of assignment of some task to specific resource:

$$U=F+Q.$$

In this expression F is the measure of unbalance and shows how balanced is the use of system resources (computational and network channels) by particular task, and Q is the measure of system resources utilization.

Fig 6b illustrates the performance of developed algorithm (bold line) comparing with random and round-robin schedulers (thin lines).

7 Conclusions

Nowadays, there is a strong interest of scientific communities from different domains in the development of distributed systems for complex problems solving with the use of high-performance computing. Grid represents an appropriate technology that enables integration and management of geographically distributed informational and computational resources. In the last years leading organizations of the NASU are involved in the research and development of Grid-based computing systems, and the first results have been already achieved. In the near future it is planned to integrate Ukrainian resources in a single infrastructure based on the high-speed network. And this infrastructure should be based on recent developments in Grid technology, high-speed networks, and multi-processors platforms.

The Grid infrastructure for satellite data processing that has been developed in the Space Research Institute of NASU-NSAU will become Ukrainian segment of the GEOSS/GMES system.

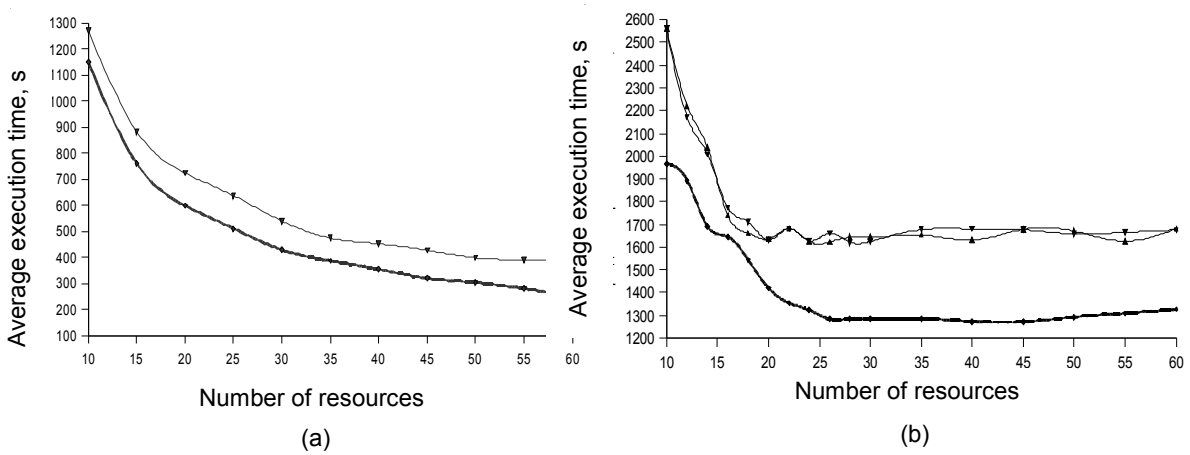


Fig. 6. Average task execution time for independent tasks (a) and data-sharing tasks (b) depending on number of resources in Grid system

Acknowledgments

This research is supported by INTAS-CNES-NSAU project “Data Fusion Grid Infrastructure”, Ref. No 06-1000024-9154, NASU Innovative Project “Development of pilot version of information infrastructure of Ukrainian segment of GEOSS” and NASU grant for Young Scintists “Development of intelligent methods and information technologies for parametric identification of hydrometeorological models”.

Bibliography

[1] Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Int. J. of High Performance Computing Applications 15 (3) (2001) 200-222.

[2] Gentzsch, W.: Special issue on metacomputing: From workstation clusters to internet computing. Future Generation Computer Systems 15 (1999).

[3] Japan Aerospace eXploration Agency (JAXA), http://www.jaxa.jp/index_e.html.

[4] Information Power Grid (IPG), http://www.ipg.nasa.gov.

[5] National Polar-orbiting Operational Environmental Satellite System (NPOESS), http://www.ipo.noaa.gov.

[6] Karajan workflow description language, http://wiki.cogkit.org/index.php/Java_CoG_Kit_Karajan_Workflow_Reference_Manual.

[7] Kussul, N., Shelestov, A., Phuong, N., Korbakov, M., Kravchenko, A. : Parallel Markovian Approach to the Problem of Cloud Mask Extraction. In: Proc. of XI-th International Conference “Knowledge-Dialog-Solution”, Varna, Bulgaria. (2005) pp. 567-569.

[8] Nguyen, T.P.: Concurrent Algorithm For Filtering Impulse Noise On Satellite Images. In: Proc. Int. Conference «Knowledge–Dialogue–Solution» (KDS–2005) (2005) 465–472.

[9] Weather Research & Forecasting model, http://wrf-model.org.

[10] University of Minnesota MapServer., http://mapserver.gis.umn.edu.

[11] OGC Standards, http://www.opengeospatial.org/specs/?page=specs.

[12] Buyya, R., Murshed, M.: GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, Concurrency and Computation: Practice and Experience (CCPE), Volume 14 Issue 13-15 (2002) 1175-1220. Wiley Press, USA.

[13] Song, H., Liu, X., Jakobsen, D., Bhagwan, R., Zhang, X., Taura, K., Chien, A.: The MicroGrid: A Scientific Tool for Modeling Computational Grids, Proc. of IEEE Supercomputing (SC 2000), Dallas, USA, (2000).

[14] Casanova, H.: Simgrid: A Toolkit for the Simulation of Application Scheduling, Proc. of the First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), Brisbane, Australia, IEEE Computer Society Press, USA (2001).

[15] Baumgartner, K., Wah, B.W.: Computer Scheduling Algorithms: Past, Present and Future, Information Sciences, vol. 57 & 58, (1991) 319-345. Elsevier Science, Pub. Co., Inc., New York, NY.

[16] Flynn Hummel, S., Schmidt, J., Uma, R.N., Wein, J.: Load-Sharing in Heterogeneous Systems via Weighted Factoring. In: Proc. of the 8th Symposium on Parallel Algorithms and Architectures (1997).

Authors' Information

Nataliia Kussul – Professor, Senior Researcher, e-mail: inform@ikd.kiev.ua

Andrii Shelestov – PhD, Senior Researcher, e-mail: inform@ikd.kiev.ua

Mykhailo Korbakov – Research Assistant, e-mail: inform@ikd.kiev.ua

Oleksii Kravchenko – Research Assistant, e-mail: inform@ikd.kiev.ua

Serhiy Skakun - PhD, Research Assistant, e-mail: inform@ikd.kiev.ua

Mykola Ilin – e-mail: inform@ikd.kiev.ua

Alina Rudakova – Research Assistant, e-mail: inform@ikd.kiev.ua

Volodymyr Pasechnik – e-mail: inform@ikd.kiev.ua

Department of Space Information Technologies and Systems, Space Research Institute of NASU-NSAU, Glushkov Ave 40, Kyiv-187, 03650 Ukraine.

DISTRIBUTED VISUALIZATION SYSTEMS IN REMOTE SENSING DATA PROCESSING GRID

Andrii Shelestov, Oleksiy Kravchenko, Mykola Ilin

Abstract: Implementation of GEOSS/GMES initiative requires creation and integration of service providers, most of which provide geospatial data output from Grid system to interactive user. In this paper approaches of DOS-centers (service providers) integration used in Ukrainian segment of GEOSS/GMES will be considered and template solutions for geospatial data visualization subsystems will be suggested. Developed patterns are implemented in DOS center of Space Research Institute of National Academy of Science of Ukraine and National Space Agency of Ukraine (NASU-NSAU).

Keywords: data visualization.

ACM Classification Keywords: I.3.2 Graphics Systems - Distributed/network graphics, C.5.0 Computer system implementation – General.

1 Introduction

Grid systems providing geospatial data are common and usually have complex visualization subsystems. Wide class of typical problems are weather prediction, satellite data processing can be solved in these systems, some of them are solved in DOS center of Space Research Institute of National Academy of NASU-NSAU. Different interfaces and architecture assumptions can make these Grid systems very hard for development and usage, lowering their value as the data source for decision making. Implementation of standards for data visualization, creation of common template solutions will simplify development and increase usability of these systems.

These approaches are used to implement distributed geospatial data visualization subsystem of national Ukrainian Earth Observation system which is developed in the frame of international program GEOSS and European program GMES.

International GEOSS program (Global Earth Observation System of Systems) is emerged to integrate national and regional Earth Observation systems [1]. One of such systems is developed within European GMES (Global Monitoring for Environment and Security) initiative. This initiative is supported by European Commission and European Space Agency and targeting on providing information services for decision making [2].

The overall structure of Ukrainian segment of GEOSS/GMES has three organizational levels. The top level is responsible for the overall management of the system, the second is responsible for integration of efforts in particular sectors of economy. At the lowest level of system's hierarchy DOS (Delivery of Service) centers are located. These centers are responsible for delivering particular services to end users [3].

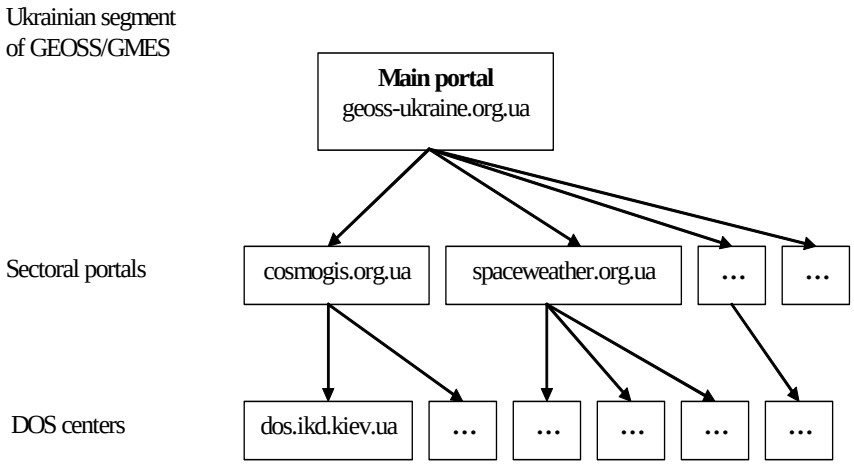


Fig. 1. Hierarchical structure of Ukrainian segment of GEOSS/GMES

To represent activities at all levels of Ukrainian segment of GEOSS/GMES the following hierarchy of Web-resources is created (Fig. 1):

- Main portal [4]
- Sectoral portals (http://cosmogis.org.ua, http://spaceweather.org.ua, …)
- Web-resource of DOS-centers

The most developed sectoral system of Ukrainian segment of GEOSS/GMES is CosmoGIS system supported by NSAU [5]. CosmoGIS is created to stimulate cooperation in the field of remote-sensing data processing and to provide end users with new quality thematic products. Environmental monitoring using remote-sensing involves execution of complex workflows of data processing and often requires computationally intensive ecological simulations. For such applications Grid computing is desirable. Typical Web-site of DOS-center presents an interface to target Grid system and provides facilities to visualize and distribute results of processing.

Contrary to top level of Ukrainian segment of GEOSS/GMES sectoral level involves substantial interactions between components (DOS-centers). One of the goal of CosmoGIS as sectoral system consists in integration of DOS-centers, in particular providing a means for distributed data visualization and delivery. To attain this goal CosmoGIS uses open standards of geospatial data presentation. At present the most advanced standards in this area both in capabilities and available software are standards of Open Geospatial Consortium (OGC). The utilizing of OGC standards ensures possibility of integration with similar systems at national level, in particular within GMES program.

In this paper different approaches to DOS-centers integration will be considered and template solutions for geospatial data visualization subsystems will be suggested. Developed patterns are used to implement DOS-center of Space Research Institute of NASU-NSAU.

2 Approaches to organization of visualization systems

One of the main obstacles on the creation of distributed systems in Ukraine is a not sufficient high throughput networks and nonuniform distribution. To account the insufficiency of high throughput networks centralized and decentralized approaches to create of distributed visualization system for geospatial data are considered [7]. The differences between these approaches consist in different traffic routing schemes between end user and DOS-centers and places where mapping products are created. Both schemes assume that DOS-centers are using OGC Web Feature Service (WFS) [8] standard to distribute vector geospatial data and OGC Web Coverage Service (WCS) [9] to distribute raster data.

The typical structure of centralized version of the system is shown on Fig. 2a. The figure shows how the thin client accesses the portal with the directory of available services and makes a request to the specified service. This request routed to the mapping service, packed into WFS/WCS request and send to the service site (DOS).

The result is routed back, processed by to the cartographical service and send to client. In this case the centralized mapping service is responsible for producing cartographical output.

This first approach exhibits ability to use thin clients (and as result to serve broader range of end users), to produce high quality mapping output independent of end clients capabilities. As a drawback this scheme has potential bottlenecks in network throughput and computational power of central mapping server.

Within decentralized scheme each DOS-center uses own mapping service (Fig. 2b). Cartographical output of service center is delivered using OGS Web Map Service (WMS) [10] protocol and client software is responsible for combining created maps. Central portal only holds references to DOS-centers and routes user requests to DOS mapping services. The second approach relaxes requirements on network throughput and available computational power at the cost of using more sophisticated clients.

In both schemes dedicated software such as geoinformational systems (GIS) can call DOS directly using WCF/WFS protocols.

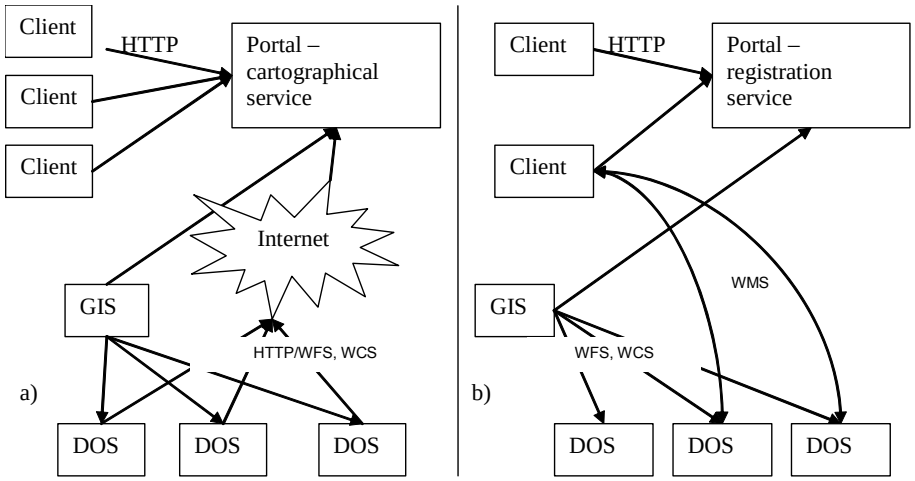


Fig. 2. (a) centralised and (b) decentralized approaches to distributed visualization of geospatial data

3 Template solutions for visualization

In this section two template solutions for visualization subsystems will be described. One solution is based on the thin client model while another utilize thick clients. Templates can be used to develop data visualization subsystems for DOS-centers and for portals at sectoral level. Both template solutions have different advantages and drawbacks. Visualization template using thin client model has a low scalability, minimalistic user interface (without navigation, dynamical scalability, etc), but does not require expensive hardware for both client and visualization server. Thick client

model visualization template has better scalability and usability, but requires expensive server or group of servers for mapping service.

3.1 Visualization systems using thin client model

First pattern of visualization system is based on the thin client model. To access the service a simple web browser is sufficient. Within this pattern, visualization system implements open standards of data presentation including OGC WMS to deliver cartographic products and OGS WFS/WCS to deliver geospatial data. Typical structure of such pattern is shown on Fig. 3

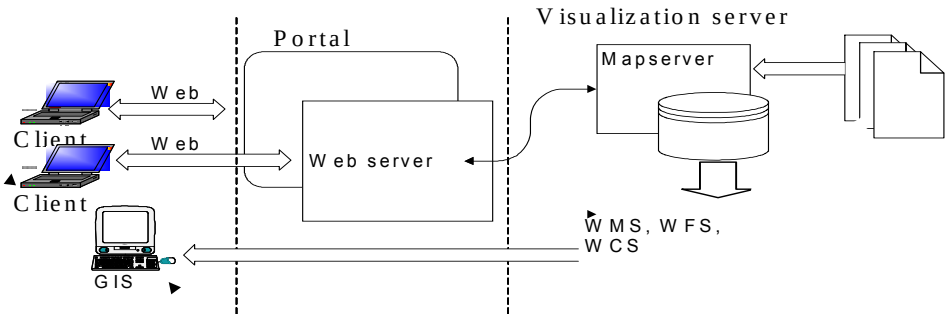


Fig. 3. Visualization system using thin client

The vector and raster data produced by Grid system is visualized by mapping service. Both mapping service and Web-interface are implemented in the framework of open source software UMN Mapserver [11]. Mapping service and visualization system located on single server, this server has sufficient performance for visualization of only a few layers on the target map. Performance restrictions are critical for visualization tasks, not all available Grid systems can use this pattern because of these restrictions. On the other side simple client software, cheaper hardware for both client and server makes this pattern very suitable in Ukrainian segment of GEOSS/GMES.

The main advantage of described pattern is the standard interface of mapping service, which grants compatibility with existing and new client applications. Once developed, client applications and DOS-centers can easily switch data sources with minor or no modifications of source code. This improves scalability and availability of entire segment. GIS can use visualization server as a client, visualization server implements standard protocols for data representation.

A typical example of thin client pattern implementation is cloud mask visualization service which is described in section 4.1.

3.2 Visualization systems using thick client model

The second pattern of visualization system is based on thick client model. To access a service a Web-browser with JavaScript is required. Key feature of proposed solution is extensibility, new versions of framework software requires more system resources. This pattern uses previous as base and adds advanced navigation capabilities for interactive users.

Another advantage of this architecture is the possibility of parallel processing of user requests allowing integration of different data sources. This feature increases the system scalability while the system remains transparent for target users. More sophisticated applications can be developed, because of performance increase, routing capabilities and load balancing. Within this template, the solution visualization subsystem is developed using open source software Cartoweb 3.2.0 [12]. In this system SOAP is used for interserver communication (among different visualization systems), allowing integration with virtually any DOS-center, even without WCS/WFS support. CartoWeb can be extended using plugin approach that makes interface modifications simple.

The main features/advantages of common Cartoweb-based interface are visible on main map control – it has scale and position arrows that can be used for scale the adjustment and navigation, this feature being commonly used with dynamic (i.e. clickable) keymap. Other commonly used features are the measuring tools for distances and surfaces measurement. The control panel has layers tree – CartoWeb supports an arbitrarily complex hierarchy of layers, with infinite depth. The interface contains a geographic query tool which can be used for geographical search. Additional features are language switch for internationalization support, users and roles support for an implementation of basic (file-based) authentication mechanism, print dialog for fully configurable PDF document production.

Typical structure of such system is shown on Fig. 4. All capabilities of previous pattern are preserved, to use new system little or no modification in existing application is required.

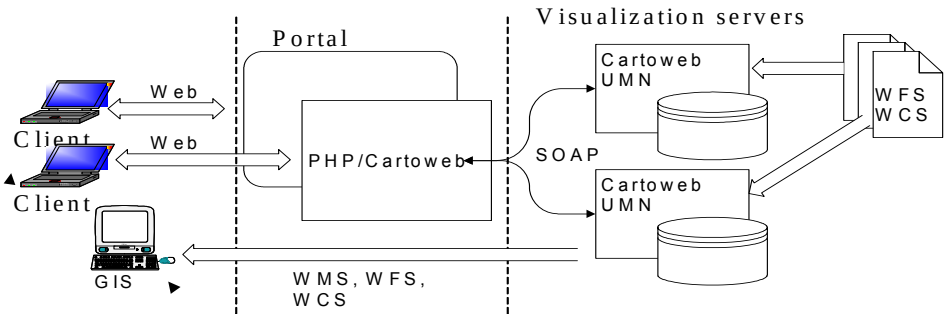


Fig. 4. Visualization system using thick client

4 Implementation

The developed patterns for visualization subsystem were used to implement the DOS-center of Space Research Institute's of NASU-NSAU [13]. This center was created under the umbrella of CosmoGIS sectoral system of Ukrainian segment of GEOSS/GMES.

The thin client-based pattern was used to implement cloud mask visualization service which is described in section 4.1, the thick client-based approach is demonstrated on example of visualization in Numerical Weather Prediction (NWP) described in section 4.2.

4.1 Service of cloud mask extraction from Meteosat remote sensing data

The cloud mask visualization is a typical application of the thin client pattern. The main source of data for this service is provided by European Meteosat meteorological satellite. The cloud mask extracted from Meteosat infrared band using Markov Random Fields (MRF) approach. Cloud mask extraction is being executed on the top of Grid system developed in Space Research Institute [14, 15].

There are few layers visualized on single machine with single service, single server has sufficient performance for smooth presentation with reasonable delays. Typical service interface is shown on Fig. 5. On the left map of this figure cloud mask is shown with country names and boundaries. Middle reference map is used for navigation. Right control panel is used for map navigation, selection of the date and additional features to be included in resulting map. Available data layers include country boundaries, Digital Elevation Model data, Meteosat infrared remote sensing data, cloud mask and clouds borders.

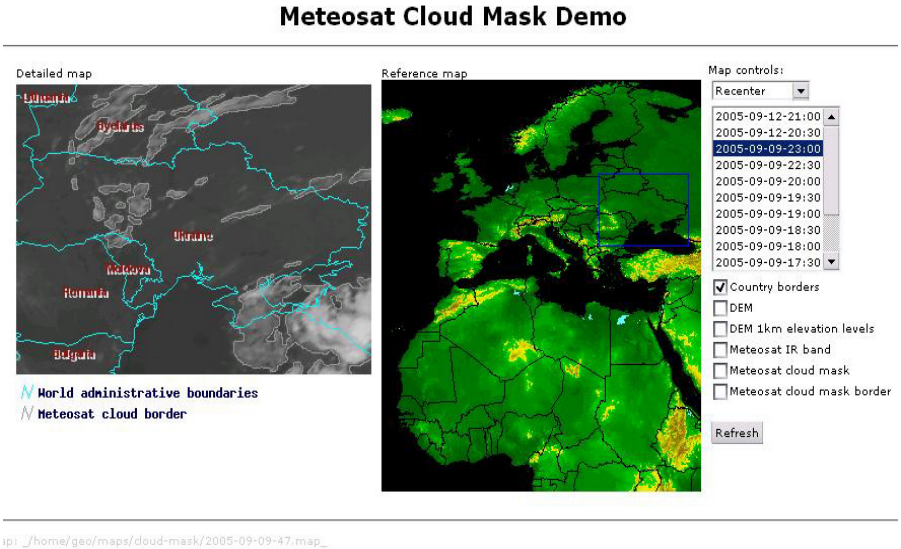


Fig. 5. Thin client system for cloud mask visualization

4.2 NWP model visualization

The visualization of NWP results is a good example of implementation of thick client-based pattern. This pattern was used to visualize results of WRF mesoscale model simulations which is regularly preformed in Space Research Institute. NWP models predict a lot of meteorological parameters. Due to the fact that these many visualization layers have to be calculated on different servers, layer options are too complex for single mapping service.

A typical user-friendly output of visualization system is shown in the Fig. 6. The main visualization controls and options located in right panel of the figure. Currently visible tab folder shows background and geopolitical reference options. Other tabs of this panel includes options to access measurement tools, which can be used for area and length calculation, print dialog for PDF exporting support, query tools for visual MapServer query creation, and outlining tools for map annotation. On the right part of the figure WRF model temperature output is shown. In the upper left corner of this map reference keymap with relief data is shown. Visualization map has advanced zooming and navigation capabilities. Zooming can be used not only with scale switch in bottom right corner, but can be applied for user-selected region. Different line and polygon drawing tools in upper panel can also be used for map annotation.

This example refers to a thick client model because navigation interface is implemented using JavaScript, uses AJAX technology for map operations.

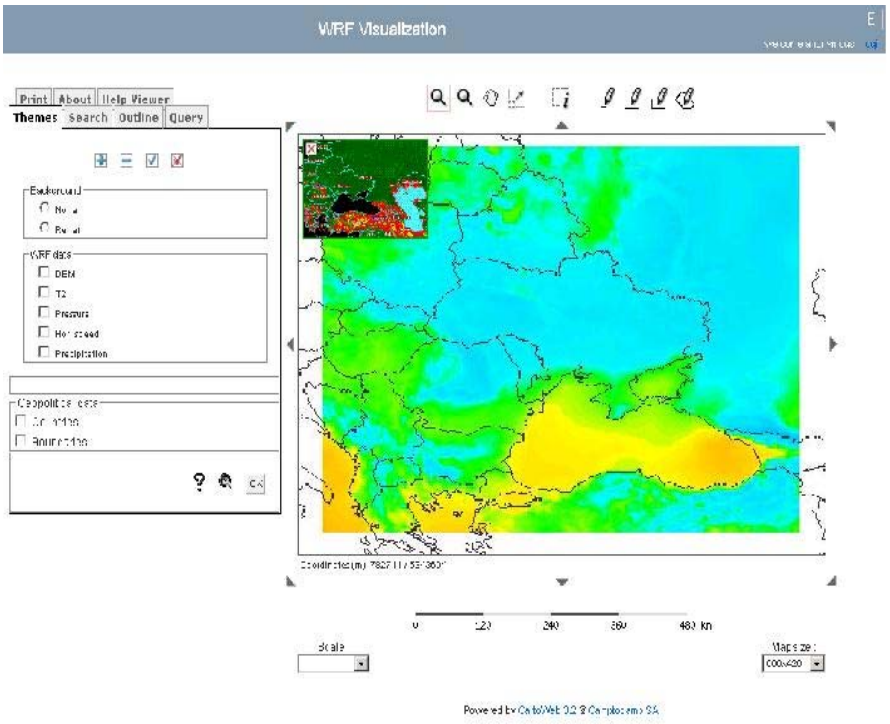


Fig. 6. Thick client system for WRF visualization

Acknowledgments

This research is supported by INTAS-CNES-NSAU project “Data Fusion Grid Infrastructure”, Ref. No 06-1000024-9154 and NASU Innovative Project “Development of pilot version of information infrastructure of Ukrainian segment of GEOSS”.

Bibliography

[1] Global Earth Observation System of Systems, http://www.epa.gov/geoss.

[2] Global Monitoring for Environment and Security, http://www.gmes.info.

[3] Fedorov, O.P., Kussul, N.N., Shelestov, A.Yu.: Problems and Prospects of Development of an Information Earth Observation System in the Ukraine (in russian). Journal of Automation and Information Sciences. Vol. 37, Issue 12, pp. 35-39. (2005).

[4] Ukrainian segment of GEOSS, http://geoss-ukraine.org.ua.

[5] Sectoral system CosmoGIS, http://cosmogis.org.ua.

[6] Space Research Institute's of NASU-NSAU service site, http://dos.ikd.kiev.ua.

[7] Kravchenko, O.M., Shelestov, A.U.: Using implementation of OGC standards to develop distributed systems for geospatial data visualization and delivery (in russian). Problems of programming. №2-3, pp. 135-139. (2006).

[8] WFS specification, http://www.opengis.org/techno/specs/02-058.pdf.

[9] WCS specification, http://www.opengis.org/techno/specs.

[10] WMS 1.1.1 specification, http://www.opengis.org/docs/01-068r2.pdf.

[11] UMN MapServer, http://mapserver.gis.umn.edu.

[12] Cartoweb, http://www.cartoweb.org.

[13] DOS-center of Space Research Institute of NASU-NSAU, http://dos.ikd.kiev.ua.

[14] Shelestov, A., Lobunets, A., Korbakov, M.: Grid-enabling Satellite Image Archive Prototype for UA Space Grid Testbed. International Journal "Information Theories and Applications. Volume 12, Number 4, pp. 351-357. (2006).

[15] Shelestov, A.Ju., Kussul, N.N., Skakun, S.V: Grid-infrastructure simulation. Problems of programming.Vol. №2-3, pp. 221-230. ISSN1727-4907. (2006).

Authors' Information

A. Yu. Shelestov – PhD, Senior Researcher, e-mail: inform@ikd.kiev.ua.

O. M. Kravchenko – PhD student, e-mail: inform@ikd.kiev.ua.

M. I. Ilin – Junior Researcher, e-mail: inform@ikd.kiev.ua.

Department of Space Information Technologies and Systems, Space Research Institute of NASU-NSAU, Glushkov Ave 40, Kyiv-187, 03650 Ukraine,

INFORMATION SUPPLY OF GEO-INFORMATION SYSTEMS FOR THE FORECASTING PROBLEM OF THE AVALANCHE DANGER

Alexander Kuzemin, Olesya Dyachenko, Darya Fastova

Abstract: This article is dedicated to the vital problem of the creation of GIS-systems for the monitoring, prognostication and control of technogenic natural catastrophes. The decrease of risks, the protection of economic objects, averting the human victims, caused by the dynamism of avalanche centers, depends on the effectiveness of the prognostication procedures of avalanche danger used. In the article the structure of a prognostication subsystem information input is developed and the technology for the complex forecast of avalanche-prone situations is proposed.

Keywords: GIS, prognostication, risk, situation, the avalanche danger

Introduction

A study of the natural calamities mechanisms, the development of their connections with the climatic and ecological changes led to the development of the new specialized systems technology for control, which was called geo-information systems (GIS). The basic tasks of GIS-systems are the development of the prognostication methodology for technogenic catastrophes, the estimation of risks and creation of decision making support systems. GIS-systems are intended for working and analysis of the enormous massifs of data for the definition of the characteristics of the zones of high risk, improvement in the planning, which precedes calamities and estimation of damage [1, 2]. The methods and techniques of GIS-systems make possible to evaluate strategies of reduction in the probability of the catastrophes occurrence, including the calculation of social and economic nature. This includes monitoring physical, biological and chemical parameters on the spot of calamity, control of measurement data and development of short- and extended forecast models. The developed GIS-systems make possible to continuously accumulate meteorological information, to perform different

calculations, to reveal regularities, to achieve a three-dimensional tying of results. The purpose of this article consists in effectiveness increase in solution taken by GIS-system due to the development of the complex forecast technology of avalanche danger. According to the stated goal, it is necessary to solve the following subtasks:

1. to determine structure, tasks and purposes of the developed GIS-system;
2. to develop structure and method of prognostication subsystem operations;
3. to determine information supply of prognostication subsystem;
4. to build the technology of the avalanche danger complex forecast.

Structure and the task of GIS-system

GIS-system is the totality of the following elements (Fig. 1):

1. *monitoring subsystem*, intended for guaranteeing of dynamic monitoring and mapping the indices of dangerous situations on that investigated of territories in the visual (tabular, graphic, cartographic and animated) form.
2. *integration subsystem* of the different data sources for the solution of the problems of control of natural catastrophic situations.
3. *analytical subsystem*, which ensures the complex analytical processing of information for the solution of complex analytical problems.
4. *prognostication subsystem*, which ensures the multivariant scenic and purposeful prognostication of situations on the base of the complex of the interconnected models of the separate parameters.
5. *system of decision making support*, based on the development of models and methods for making of adequate decisions of operational and strategic nature.
6. *subsystem of results representation*, which ensures data presentation in the most visual tabular, graphic and cartographic form, which reflects qualitative characteristics and basic tendencies of the indices of situation.

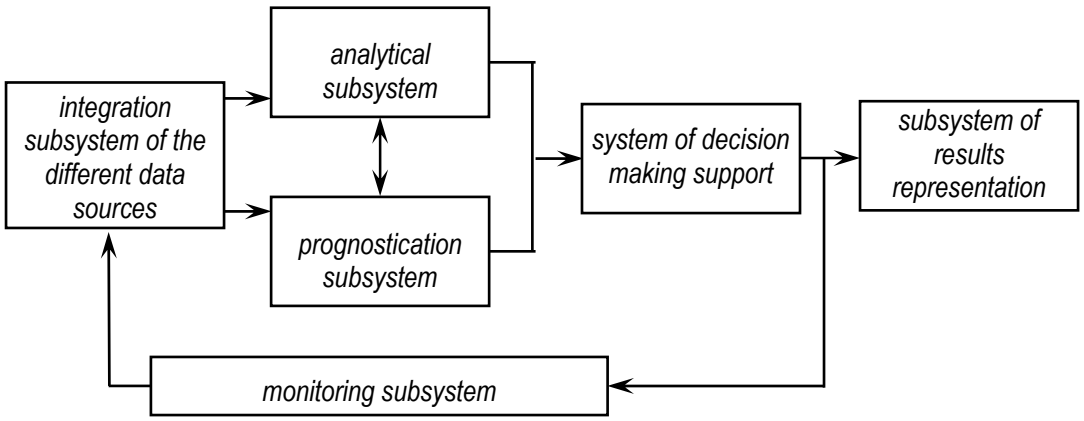


Fig. 1. GIS-system structure.

The distinctive special feature of geo-information systems is the feedback, whose function fulfills the subsystem of monitoring. In general form the role of GIS-technologies in avalanche studies is reduced to the synthesis of knowledge about the relief, the climate and the preceding events, for the purpose of possibility determination of gathering snow avalanches. For this in the GIS environment are imported already existing maps or new projects are created. The analysis of the works, dedicated to use GIS in avalanche studies, showed that the GIS-technologies at present adapt for the solution of the following problems:

- development of the zones of the origin of avalanches;
- simulation of processes and phenomena, which determine the conditions for gathering snow avalanches;
- definition of lethal areas;
- creation of the cadastral surveys of avalanche centers, data bases about the avalanches;
- forecast of avalanche danger.

Subsystem of prognostication

The technology of the prognostication of avalanche danger is the information complex, which consists of three basic blocks (Fig. 2):

1. database - is intended for collection, storage and initial processing of the data of hydrographic and weather services, snow-avalanche stations and electronic charts of surface of locality, which contain information about snow accumulations, to the underlying surface and so forth
2. Mathematical and algorithmic guarantee - is the collection of mathematical methods and approaches, on base of which is produced the simulation and the prognostication of avalanche-prone danger. The prognostication of avalanche-prone situation is characterized by four basic parameters: by place, by type, by time and with its degree of power. Each of the characteristics has available their mathematical, algorithmic and program apparatuses.
3. block of results assignment for different levels of users - the obtained forecasts are analyzed by experts and leaders of Emergency And Disaster Relief Ministry, after which they are transferred for modification to the system of decision making support for the purpose of use with the correction of anti-avalanche measures and to the elimination of the consequences of gathering avalanche.

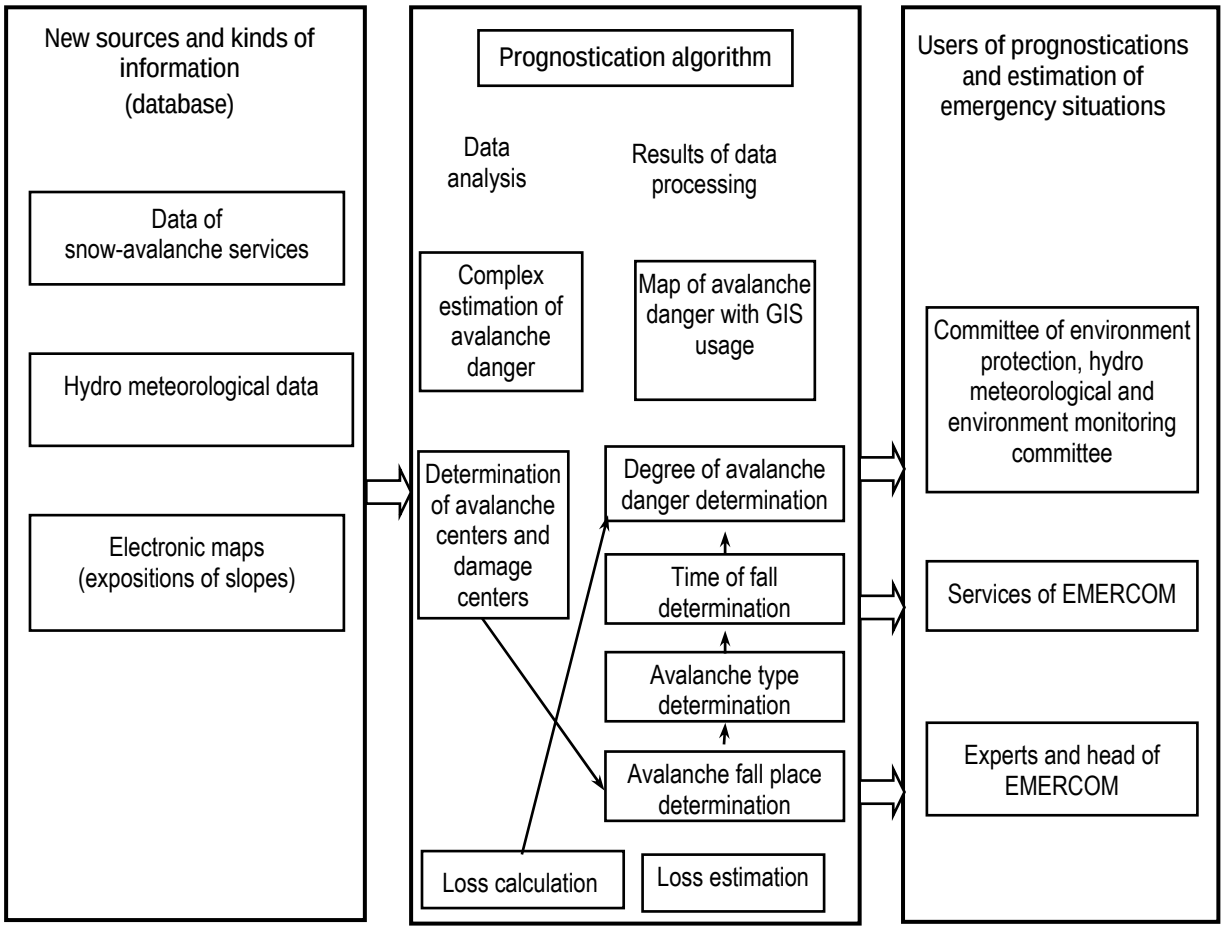


Fig. 2. Technology of avalanche danger prognostication.

For the realization of the complex forecast of avalanche-prone situations it is necessary to determine the place of gathering avalanche, to produce its classification on the genesis, to determine time and degree of the power of avalanche. The place of gathering avalanches is determined with the aid of the electronic charts of area relief, on which is determined the slope of slopes, they calculate the thickness of snow cover and the distance of ejection.

The genetic type of avalanche depends on the geographical and climatic special features of mountain locality. 5 basic genetic types of avalanches can be specified:

1. avalanche of the freshly fallen snow.
2. wet avalanches.
3. avalanche of snowstorm snow.
4. avalanche of temperature reduction.
5. avalanche of sublimation diaphthoresis.

This classification of avalanches is conditional, since it is very difficult to find the relation of the processes of the appearance of avalanches with the natural conditions of avalanche-prone regions. Depending on the genesis of avalanches the procedures of remaining characteristics prognostication of avalanche danger are selected. In the world practice a large quantity of power estimation scales of avalanches is used - these are the European scale, the American scale, the French scale of avalanches power evaluation [4]. In the developed subsystem of the prognostication of avalanche danger adapts the French scale of the evaluation of the power of avalanche danger, developed By F. Rapin [5] (table 1). The hazard level is evaluated by five progressively growing steps, which are described through such physical parameters as the damaged territory, the thickness of avalanche board, the volume of derailed snow, dynamic pressure, probability of caving avalanches and their effect on the vital activity in the mountains.

Table 1.

Scale of the avalanches power degrees

	Degree of avalanche power	Physical parameters	Probability of avalanche caving
1	Insignificant	Damaged territory:~0.2 GA Thickness of the avalanche board: 20 cm Amount of derailed snow: $\sim 100\,\mathrm{m}^3$ Dynamic pressure: ~2 KPa	Caving is possible only with the very significant increment loads on the separate very steep slopes. Spontaneously can occur only the motions of snow
2	Moderate	Damaged territory:~1 GA Thickness of the avalanche board: 40 cm Amount of derailed snow: $\sim 1000\,\mathrm{m}^3$ Dynamic pressure: ~10 KPa	Caving is possible with the significant increment loads first of all on the slopes indicated, spontaneous caving of avalanches highly improbably
3	Significant	Damaged territory:~5 GA Thickness of the avalanche board: 80 cm Amount of derailed snow: $\sim 10\,000\,\mathrm{m}^3$ Dynamic pressure: ~50 KPa	Caving is possible with the insignificant increment load on the slopes indicated. Is possible caving separate average and less probably large avalanches
4	Large	Damaged territory:~20 GA Thickness of the avalanche board: 150 cm Amount of derailed snow: $\sim 80\,000\,\mathrm{m}^3$ Dynamic pressure: ~200 KPa	Caving is possible on the majority of slopes with insignificant increment load
5	Very large	Damaged territory:~50 GA Thickness of the avalanche board: 250 cm Amount of derailed snow: $\geq 400\,000\,\mathrm{m}^3$ Dynamic pressure: ~500 KPa	A numerous spontaneous avalanche caving on any slopes are expected

A time aspect of avalanche danger forecast provides for determination of the possibility of gathering avalanches in the assigned territory into the caused time interval. Among the difficulties, connected with the calculation of the time aspect of the avalanche activity, it is possible to state [6]:

1. Provided in the scientific literature classification of forecasts on short -, middle- and long-term does not use the fixed time intervals for their separation. The analysis of works on the prognostication of avalanche danger shows that in practice the forecast can be comprised for day, 48 hours, 72 hours, for the winter season, for the long-standing interval of time.
2. Forecasts of avalanche danger are created with the use of those of specially developed for the region or the separate center procedures, which determine the algorithm of avalanche danger detection.
3. A lot of procedures provides for the forecast of avalanche-prone period - the time interval, for which it will remain the action of avalanche formation factor. Usually, this approach is used with the forecast of avalanches during the snowfalls and snow-storms. Avalanches are forecasted from the moment of achieving the critical conditions to the end of the snowfall (snow-storm), and for the period from one to two days from their end - thus far the instability of snow cover remains.

Lead time (time between forecast composition and beginning of its action) of forecast, placed in many procedures of forecast is equal to zero. In practice this indicates the statement of facts of reaching the avalanches of conditions critical for the gathering. The basic reasons for this lie in the transience of avalanche-prone situation appearance (from several hours to days), a constant change in the meteorological conditions, impossibility of the continuous and general collection of necessary information. The complex forecast of avalanche danger is the necessary information, which is the basic tasks of the system of support and decision making are:

1. guarantee of the planning, planning, controlling organizations with the information about propagation of natural dangers, creation of land cadastre, selection of optimum places for building of linear and area units (Russia, USA, Switzerland, Austria and other.);
2. ecological control of region - influence of avalanches on the dynamics of landscapes, the nature and the boundaries of plant communities;
3. selection of tourist groups safe movement ways;
4. development of anti-avalanche activities;
5. study of interrelations of dangerous natural and anthropogenic phenomena (Russia, The USA).

Conclusions

This article examines the geo-information system, intended for predicting of avalanche danger and decision making by the averting and overcoming of its consequences. Structure and tasks of the developed GIS-system are examined. The subsystem of prognostication is in detail represented, is described its information input, which consists of the data base, mathematical and algorithmic complexes, the block of the assignment of results. Is proposed the technology of the complex forecast of the avalanche danger, which includes the determination of the position of gathering, the classification of avalanches on the genesis, the determination of time and degree of the power of avalanches.

Bibliography

1. Kupcova A.V., Perekrest V.V.. GIS of Kabardino-Balkar republic created and working. Information bulletin of GIS Association. Moscow, 1996, № 3(5), p.24-25 (RUS)
2. Pertziger, F. 1998. Using of GIS technology for avalanche hazard mapping, scale 1:10 000. NGI, Oslo, pub. Nr.203, 210-214.
3. Bozhinskiy A.N., Losev K.S. General avalanche-caring. – Leningrad.: Hydrometeoizdat, 1987. – 280 p. (RUS)
4. Buser O., Fuhn, P., Gubler W., Salm B. Different methods for the assessment of avalanche danger. Cold. Reg. Sci. Technol., 1985, 10 (3), 199-218.
5. Rapin F. A new scale for avalanche intensity. International Snow Science Workshop., 2002, vol.2, 103-110
6. Fuhn P. An overview of avalanche forecasting models and methods. Oslo, NGI, Pub.N 203, 1998, 19-27.

Authors' Information

Alexander Ya. Kuzemin – Prof. of Information Department, Kharkov National University of Radio Electronics, Head of IMD, Ukraine, e-mail: kuzy@kture.kharkov.ua

Olesya Dyachenko – phd student, Kharkov National University of Radio Electronics, Ukraine

Darya Fastova – phd student, Kharkov National University of Radio Electronics, Ukraine

DATA INDEPENDENCE IN THE MULTI-DIMENSIONAL NUMBERED INFORMATION SPACES

Krassimir Markov

Abstract: The concept of data independence designates the techniques that allow data to be changed without affecting the applications that process it. The different structures of the information bases require corresponded tools for supporting data independence. A kind of information bases (the Multi-dimensional Numbered Information Spaces) are pointed in the paper. The data independence in such information bases is discussed.

Keywords: Data independence; Multi-dimensional Numbered Information Spaces

ACM Classification Keywords: D.2 Software Engineering; H.2 Database Management

Introduction

It is well-known, the concept of the data independence denotes that a database is designed and maintained independently of applications that retrieve and manipulate the data [CODASYL, 1971], [Martin, 1975], [Date, 1977], [Gabrovsky and Markov, 1977], [Connolly and Begg, 2005]. Dr. E.F. Codd had published a list of rules that concisely define an ideal relational database, which have provided a guideline for the design of all relational database systems ever since. The rules from 8 till 11 concern the data independence [Codd, 1985]. There are several kinds of data independence.

Physical data independence means that physical details of data organization and access are transparent for the application programmer. The physical details are determined by the database designers or even earlier, by the designers of a database management system. Some physical details (e.g. indices supporting the access to data, special file organization, special methods of performing operations, etc.) are under control of a database administrator (DBA). DBA uses special administration module to tune the database operation according to the actual demands of applications, but still, nothing in applications has to be changed due to the tuning [Subieta, 2005]. Physical data independence is the rule 8 of Codd's list - the user is isolated from the physical method of storing and retrieving information from the database. Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods.

Logical data independence means that DBA is able to perform some operations on the database structure, for instance, add new data kinds, add or remove some object or table attributes, change user privileges, add and remove views, database procedures, triggers etc. without unconscious influencing the applications [Subieta, 2005]. Logical data independence is the rule 9 of Codd's list - how a user views data should not change when the logical structure (tables structure) of the database changes. Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit unimpairment are made to the base tables. (This rule permits logical database design to be changed dynamically, e.g. by splitting or joining base tables in ways which do not entail loss of information.) Continuing this principle we may talk about *Conceptual data independence* that means that DBA is able to change the structure of the database conceptually without changing existing (legacy) applications, for instance, through special wrappers, mediators, views, updatable views, do instead of triggers, and other means that allow to change significantly the database schema and its organization, perhaps with minor changes of applications. This kind of data independence is referred to as schema evolution and conceptually is close to software change management methods and the aspect-orientation in databases. [Subieta, 2005]

Integrity Independence is the rule 10 of Codd's list - the database language (like SQL) should support constraints on user input that maintain database integrity. Integrity constraints must be definable in the relational data sub-language and storable in the catalogue, not in the applications program. Certain integrity constraints hold for every relational database, further application-specific rules may be added.

Distribution Independence is the rule 11 of Codd's list – the user should be totally unaware of whether or not the database is distributed (whether parts of the database exist in multiple locations). A relational DBMS has

distributional independence - i.e. if a distributed database is used it must be possible to execute all relational operations upon it without knowing or being constrained by the physical locations of data. This must apply both when distribution is originally introduced, and when data is redistributed [Codd, 1985].

It is clear; the importance of data independence in a database management system is well recognized in the database community. To ensure the data independence is the reason for developing the three levels database architecture. Its goal is to separate the user applications and the physical database. Basically, “Three-schema Architecture” has an “Internal” level, a “Conceptual” and an “External” level. The advantages of the three tiered architecture are that this division into levels allows both developers and users to work on their own levels. They do not need to know the details of the other levels AND they do not have to know anything about changes in the other levels.

The Project INSPIRE

A particularly interesting example of multi level implementation in practice is the European Union’s INSPIRE (Infrastructure for SPatial Information in Europe) initiative [INSPIRE, 2007]. This was launched in 2001 with the objective of making available relevant, harmonised and quality geographic information to support the formulation, implementation, monitoring and evaluation of Community policies with a territorial dimension or impact' (http://inspire.jrc.it). INSPIRE is seen as the first step towards a broad multi sectoral initiative which focuses on the spatial information that is required for environmental policies. It is a legal initiative that addresses “technical standards and protocols, organisation and coordination issues, data policy issues including data access and the creation and maintenance of spatial information” [Masser, 2005].

The Figure 1. provides a simplified overview of key elements in the technical architecture of INSPIRE.

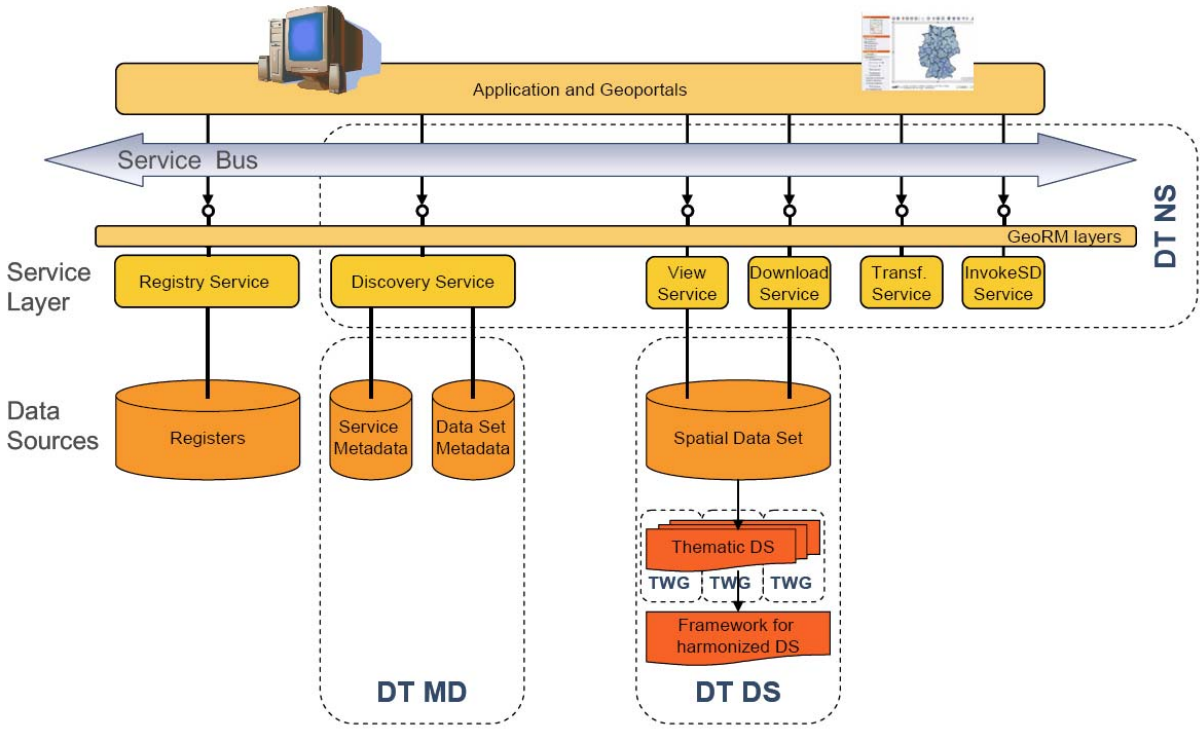


Figure 1: INSPIRE technical architecture overview presented in [INSPIRE TR, 2007].

The core resource in the diagram is the actual content, i.e. the spatial data in spatial data sets. “Spatial data” denotes any data with a direct or indirect reference to a specific location or geographic area The use of the word “spatial” in INSPIRE is unfortunate as the meaning goes beyond the meaning of “geographic” – which is understood to be the intended scope. Therefore, “spatial data” is understood as a synonym for the term

"geographic information" as used in the ISO 19100 series of International Standards. "Spatial dataset" denotes identifiable collection of spatial data.

All other resources shown in the diagram, e.g. data set metadata, are only needed to find, access, interpret or use the spatial objects in the spatial data sets that form part of the infrastructure. "Spatial object" denotes an abstract representation of a real-world phenomenon related to a specific location or geographical area. Note: This term is understood as a synonym for geographic feature as used in the ISO 19100 series of International Standards [INSPIRE, 2007].

It is important to note that in INSPIRE all access to spatial data and metadata occurs via spatial data services. The implementation platform for these services is expected to be web services. All services are described by service metadata (service descriptions), allowing humans and software applications to discover specific service instances in the infrastructure [INSPIRE TR, 2007].

The INSPIRE infrastructure will be built upon existing or emerging infrastructures in the member states and international organisations. In particular it has to be emphasised that changes to existing data capturing, updating and management processes within the member states and international organisations are in general not foreseen by the implementation of INSPIRE. Instead, the INSPIRE Implementing Rules aim at providing access to existing spatial data in a harmonised way.

In INSPIRE there are 34 different spatial data themes. For some of them already there exist specialized information bases - cadastral parcels, geographical grid systems, meteorological geographical features, etc. Only for a part of them are developed web access and good end user support. A large group of themes are under investigation and under current or near future development of the appropriate information bases. For instance, such themes are:

- Administrative units - Units of administration, dividing areas where Member States have and/or exercise jurisdictional rights, for local, regional and national governance, separated by administrative boundaries.
- Elevation - Digital elevation models for land, ice and ocean surface. Includes terrestrial elevation, bathymetry and shoreline.
- Human health and safety - Geographical distribution of dominance of pathologies (allergies, cancers, respiratory diseases, etc.), information indicating the effect on health (biomarkers, decline of fertility, epidemics) or well-being of humans (fatigue, stress, etc.) linked directly (air pollution, chemicals, depletion of the ozone layer, noise, etc.) or indirectly (food, genetically modified organisms, etc.) to the quality of the environment.
- Utility and governmental services - Includes utility facilities such as sewage, waste management, energy supply and water supply, administrative and social governmental services such as public administrations, civil protection sites, schools and hospitals.
- Environmental monitoring facilities - Location and operation of environmental monitoring facilities includes observation and measurement of emissions, of the state of environmental media and of other ecosystem parameters (biodiversity, ecological conditions of vegetation, etc.) by or on behalf of public authorities.
- Natural risk zones - Vulnerable areas characterised according to natural hazards (all atmospheric, hydrologic, seismic, volcanic and wildfire phenomena that, because of their location, severity, and frequency, have the potential to seriously affect society), e.g. floods, landslides and subsidence, avalanches, forest fires, earthquakes, volcanic eruptions.
- Atmospheric conditions - Physical conditions in the atmosphere. Includes spatial data based on measurements, on models or on a combination thereof and includes measurement locations.
- Habitats and biotopes - Geographical areas characterised by specific ecological conditions, processes, structure, and (life support) functions that physically support the organisms that live there. Includes terrestrial and aquatic areas distinguished by geographical, abiotic and biotic features, whether entirely natural or semi-natural.

  etc.

These themes are characterized by the need of collecting, storing, processing and distributing the space information with more than usual three geographical dimensions. In some cases the number of dimensions

exceeds one hundred. In addition, the collected information is not homogeneous and there exist the problem of mapping the different parts of information and easy connecting them to geographical coordinates without of permanently duplicated triples of values. For this type of practical necessity it is important to propose new tools with corresponding possibilities.

One of the main problems to be solved is representing the corresponded digitalized information in the appropriate data bases which are aimed to support time depended, multidimensional, multimodal and multimedia, individualized and confidential access to searched digitalized information objects. In the same time, the digitalized information objects need appropriate tools for storing, retrieving, processing and multimodal time depended representing for a great number of types of users. This problem could not be solved using popular in the practice (as a rule – relational) data base management systems (DBMS). The current multidimensional time depended extensions, such as these in the newest versions of the Oracle DBMS, are aimed to operate with not so complex and complicated information objects. The main area of implementing of such systems is the business information service. The pointed themes of INSPIRE pose the question about developing of principally different approach for building the information bases.

One such approach, which has been developed and experimented more than for twenty years, is using the multi-dimensional numbered information spaces [Markov, 2004]. The main its advantage is the possibility to build space hierarchies of information objects and the great power for building interconnections between information elements of the stored in the information base objects. Practically unlimited number of dimensions and the opportunity of representing and storing the information only about the existing parts of the real objects make possible the creating effective and useful tools for working with information. This approach allows possibility for building the very large information bases and supporting the time depended, multidimensional, multimodal and multimedia, individualized and confidential access to searched digitalized information objects.

Data Independence in the Multi-dimensional Numbered Information Spaces

The main peculiarity of the multi-dimensional information bases is the need of multi-dimensional schema at every level and very important feature is supporting the corresponding data independence, i.e. the capacity to change the schema at one level of the information base without having to change the schema at the next higher level. In other words, if the schema at one level is changed, the mapping to the next higher level needs to be changed to ensure the schema at the next level to remain unchanged.

The Internal Level is a description of the physical storage structure of the information base. The operations performed here are translated into modifications of the contents and structure of the files (archives).

The only tool which allows physical organization of multi-dimensional numbered information spaces is the FOI Archive Manager (ArM)®. ArM is based on the "Multi-Domain Information Model" (MDIM) [Markov, 2004]. One of the first goals of the developing of ArM was representing the digitalized military defense situation which is characterized with variety and complexity of objects and events which occur in the space and time and have long period of variable existence. The great number of layers, aspects and interconnections of the real situation may be represented only by multi-dimensional information spaces.

The ArM main information structures are:

- Basic information elements - arbitrary long strings of machine codes (bytes) which may represent information structures of any kind. When it is necessary the strings may be parceled out by lines. The length of the lines may be variable.
- Numbered information spaces of different ranges - the basic elements are organized in hierarchy of the numbered information spaces with variable ranges.

Every element as well as every space has unique number in the space it belongs. This way the element may be accessed by correspond "space address" (coordinates) given via coordinate array of numbers of the spaces that contain it in the hierarchy. The quantity of levels of the hierarchy is unlimited.

So, we have only two constructs for the physical organization of the information base – basic information elements and numbered information spaces, which may be accessed exclusively via coordinates and the Archive Manager provides 100% physical data independence.

The Logical level is a description of the structure of the entire information base. It hides the details of physical storage and concentrates on describing entities, data types, relationships, user operations, and constraints. The logical organization of multi-dimensional numbered information spaces is too complicated and it is very important to have special multi-dimensional logical schema and corresponding technology for operating with it. Such special technology called Cell Oriented Programming (CellPro) was presented in [Markov et al, 1995]. The mathematical foundations of CellPro may be find in [Lisper, 1989]. The meta-information for describing the logical organization and for its mapping over the physical one is represented by the Cells and Cell Structures. It is stored in special mapping ArM-archives, i.e. again in multi-dimensional numbered information spaces. The FOI System Builder (SyB) is a tool for multiagent cell oriented programming. The SyB main features include defining the cell structures and using the different types of cells; sets of cells; cell scripts as the specific cells' configurations with fixed or variable structure and activities; sets of cell scripts etc. SyB is constructed by System Building Service Module (SyB_SM) and Real-time Management System (SyB MS).

The SyB_SM is a system for service the defining the cells, sets of cells and cell scripts. The SyB_SM can translate the descriptions of the scripts in the internal format for the SyB_MS interpretation. The SyB_MS is a system for real-time concurrent control using the cell scripts. It may be linked with a user program and may be called using a special procedure type of interconnection.

The Cells are capsulated items that have their own description and activity; i.e. the cells are agents. The description of a cell may contain: name of the cell; type of the cell; definition of the cell activity; interconnection between the cell and other cells, etc. The cell may be:

- data cell which may contain: single data (strings of any size; integer or real numbers; dates; graphic images, search patterns, scripts); sets of single data, or sets of more complex structures which may contain both sets and single data
- functional cell which may contain built-in standard functions, constructively integrated user defined functions, external standard or user defined programs.

The activity of the cell depends of its type. For instance it may be a simple operation with the standard data (such as creating, deleting, editing and copying) or more complex information processing. The cell may execute the build in functions or scripts as well as the external standard programs such as search processing or hierarchical hypertext service. It may perform compress and decompress of information subspaces, transfer the information between points of physical information space, etc. At the end the activity of the cell may be executing the user defined scripts or programs of any kind, which may be written using different programming languages.

There are two general categories of cells: user accessible and user not accessible. The first one may be used for building the user interface with given concrete application and the second - for organizing the information base and internal information processing.

In addition to standard characteristics, the user visible and/or accessible cells have theirs screen representations and relations with representations of the other cells of such type.

The cells may have two possible types of connection with the data – absolute or relative. The absolute connection is given directly by set of coordinates, organized in the coordinate array. The relative connection is given by a combination of base, offset and index coordinate arrays.

The External Level is formed by the end users' special views that are tailored to their specific needs. Some of these views may be forms to fill out, others for interactive retrieval of information, etc.

Each user group refers only to its own external schema so the DBMS must transform a request on an external schema into a request against the conceptual schema, then into a request on the internal schema for processing. The process of transforming requests and results between levels are called mappings. These mappings may be time consuming, so some many DBMS do not separate the three levels completely.

In multi-dimensional information bases the mappings could not be simple functions. In addition the dynamic of user activities needs special tools for mapping the user views to the schema. In this work such mappings are realized using the SyB's scripts [Markov et al, 1999].

The SyB's scripts are specific cells' configurations with fixed or variable structure. It is possible to describe different configurations (not only the spreadsheets type) of the cells.

The fixed script's structure may be created and used as it was defined without any change during the usage. The variable one may be changed during the usage in accordance with any conditions. The variable structure may be connected to a special index of co-ordinates for building the final configuration of the cells in a given script.

There are several cell types used only for fixed cell scripts: the text string <S>, the integer <I> and real <R> number, the script <T>, the user program <P>, the standard function <F>, the read only cell <L>, etc.

The cell types used both for fixed and variable cell scripts are string text cell <Z>, number cell <X> or <Y>, date cell <D>, and functional cell <W>, etc.

The difference between cell's types is in the possibility for iterative access to the information space, which is available for variable cell scripts. This means that the cell may take part in the script many times and every time it cans access different zone of the archives. As usual special kind of indexes is tool for such adaptive processing.

It is clear, the <S> and <Z> cells may contain only text strings and the cells <I>, <X>, <R> and <Y> - only numbers. The cell <D> services using the dates. The activation of <T> cell will start execution of the subscript, which the cell can contain. True the <P> cell we may start execution of the user-defined program, which is pointed in the cell as well as <F>, and <W> cells will start execution of the user-defined function, which is connected, to the cell. The possibility of the <L> cell is only to visualize its content.

The description of the script may be done using the SyB language. It contains three main parts: interconnection between cells, individual functional type of the cell and activities of the cell.

The interconnection between the cells may be given by the exactly, default or functionally described cell names or coordinates i.e. addresses. Any function for description of co-ordinates may be the built-in or the user given. The built-in function of the SyB is iterative and may be used for description of repetitively use of the cells of the CIS. For instance when one wants to describe a table where every row contains the same set of cells but with other values and activities he may use this type of the description. The user given functions may access the whole information space.

Integrity Independence in the multi-dimensional information bases needs to be cleared. In the relational DB there are primary and secondary keys but when the dimensions are more than two it is not clear what will play the role of keys and what should support constraints on user input that maintain database integrity. Integrity constraints must be definable in any sub-language and storable in the catalogue. In the frame of Cell oriented approach, the integrity is supported by implementing the constraints in the cells, in the structures of cells as well as in the cell scripts.

Distribution Independence plays a very useful role in the development of Geospatial Web Service architectures [Dadi and Di, 2007]. It is important to note that for INSPIRE it is assumed that all kind of data and metadata access and processing are performed using web services. All services are described by service descriptions (service metadata, as part of the INSPIRE metadata), allowing humans and software applications to discover specific service instances in the infrastructure and invoke them automatically [INSPIRE TR, 2007].

The INSPIRE Network Services can be seen as the protocol being used to realise a pan-European geo spatial service bus (see fig. 2):

- Different *SDI providers* who contribute INSPIRE-conforming services (access only)
- INSPIRE Network Services expose services for machine-to-machine communication. At least a workflow that follows the “publish – find – bind” design pattern should be possible. However, users do not necessarily have to follow this pattern; they can also invoke services directly.
- INSPIRE Applications solving specific tasks by involving INSPIRE services.
- INSPIRE geo-portal at Community level and further Member States access points offering INSPIRE functions to the different user groups (usage of INSPIRE services). A user can access services on an EU level via the INSPIRE geo-portal but also on a MS level – usage on the EU level offers the advantage to access data that integrates seamlessly data from different member states.

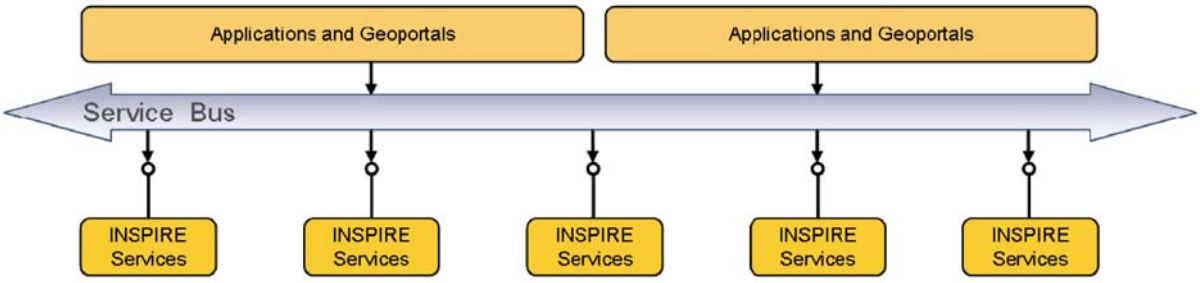


Figure 2. INSPIRE Network Service bus presented in [INSPIRE TR, 2007].

Following this understanding we may point that in our case two types of distribution independence we may have – at internal (physical) level and at the external (user) level. In the first case, the information base is assumed to be distributed and all its part are accessible automatically via INSPIRE Service bus. In the second case, the scripts may contain information request to different part of information base distributed at different servers and the integration of information will be provided by the used INSPIRE portal.

Conclusion

It is clear; the importance of data independence is well recognized in the database community. The data independence is the main direction for investigation especially in the multi-dimensional spatial information bases.

The main conclusion is that we are on the threshold of the new style of organization of information which needs special attention. The generalization and implementing of the techniques for data independence in a special kind of information bases (the Multi-dimensional Numbered Information Spaces) ware discussed in the paper.

To ensure the data independence is the reason for developing the three levels database architecture. The advantages of the three tiered architecture are that this division into levels allows both developers and users to work on their own levels. The main levels of data independence were remembered. The investigation has been based on the ideology of the European Union's INSPIRE (Infrastructure for SPatial Information in Europe) initiative [INSPIRE, 2007]. The main 34 themes of INSPIRE were discussed and it was drawn attention to a large group of themes which are under investigation and under current or near future development of the appropriate information bases. The pointed themes of INSPIRE pose the question about developing of principally different approach for building the information bases.

One such approach is using the multi-dimensional numbered information spaces [Markov, 2004]. The main its advantage is the possibility to build space hierarchies of information objects and the great power for building interconnections between information elements of the stored in the information base objects. Practically unlimited number of dimensions and the opportunity of representing and storing the information only about the existing parts of the real objects make possible the creating effective and useful tools for working with information. This approach allows possibility for building the very large information bases and supporting the time depended, multidimensional, multimodal and multimedia, individualized and confidential access to searched digitalized information objects.

The main levels of data independence in the multi-dimensional numbered information spaces were described. The main peculiarity of the multi-dimensional information bases is the need of multi-dimensional schema at every level and very important feature is supporting the corresponding data independence, i.e. the capacity to change the schema at one level of the information base without having to change the schema at the next higher level.

The logical organization of multi-dimensional numbered information spaces is too complicated and it is very important to have special multi-dimensional logical schema and corresponding technology for operating with it. Such special technology is Cell Oriented Programming (CellPro). The meta-information for describing the logical organization and for its mapping over the physical one is represented by the Cells and Cell Structures.

The external level of data independence is based on cell scripts. In multi-dimensional information bases the mappings could not be simple functions. In addition the dynamic of user activities needs special tools for mapping the user views to the schema.

Integrity Independence in the multi-dimensional information bases needs to be cleared. In the frame of Cell oriented approach, the integrity is supported by implementing the constraints in the cells, in the structures of cells as well as in the cell scripts.

Distribution Independence plays a very useful role in the development of Geospatial Web Service architectures. From point of view of the project INSPIRE were commented the web based possibilities for data independence.

Acknowledgments

This work is a part of the project "ITHEA XXI", partially financed by the Consortium FOI Bulgaria. Author is indebted to Krassimira Ivanova and Ilia Mitov for the collaboration in this project.

Bibliography

[CODASYL, 1971] Codasyl Systems Committee. Feature Analysis of Generalized Data Base Management Systems. Technical Report, May, 1971 / Информационные системы общего предназначения (Аналитический обзор систем управления базами данных). Москва, Статистика, 1975.

[Codd, 1985] E.F. Codd. "Is Your DBMS Really Relational?" and "Does Your DBMS Run By the Rules?". ComputerWorld, 14 and 21. October 1985

[Connolly and Begg, 2005] T. Connolly, C. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 4th ed., Pearson Education Ltd., 2005

[Dadi and Di, 2007] U.Dadi, L.Di. Data Independence and Geospatial WEB Services. Geoinformatics 2007 Conference (17–18 May 2007). http://gsa.confex.com/gsa/2007GE/finalprogram/abstract_122248.htm

[Date, 1977] C.J. Date. An Introduction to Database Systems. Addison-Wesley Inc. 1975. / К.Дейт. Введение в системы баз данных. Москва, Наука, 1980.

[Gabrovsky and Markov, 1977] I.Gabrovsky, Kr.Markov. About Constructing Data Independent Programs using the Package BOMP. Proceedings of the Conference of professionally connected users, Varna, 1977. pp.240-245. (in russian)

[INSPIRE TR, 2007] Infrastructure for Spatial Information in Europe Reference: INSPIRE Technical Architecture Overview, 05-11-2007 Page 5 of 12

[INSPIRE, 2007] DIRECTIVE 2007/2/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 14 March 2007, establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Official Journal of the European Union 25.4.2007 L 108/1 http://www.epsiplus.net/content/download/3477/38314/file/l_10820070425en00010014.pdf

[Lisper, 1989] B.Lisper. Syntesizing Synchronous Systems by Static Scheduling in Space-Time. Lecture Notes in Computer Science, No.: 362. Springer-Verlag, Berlin, 1989.

[Markov 2004] K. Markov. Multi-Domain Information Model. Int. Journal "Information Theories and Applications", 2004, Vol. 11, No. 4, pp. 303-308

[Markov et al, 1995] K.Markov, K.Ivanova, I.Mitov. Cell Oriented Programming. International Journal "Information Theories & Applications" (IJ ITA), 1995, Vol. 3, No. 1.

[Markov et al, 1999] K.Markov, K.Ivanova, I.Mitov. *Multiagent Information Service Based on Scripts*. Научно-теоретический журнал „Искусственный интелект", №2, 1999, ISSN 1561-5359, ИПИИ НАНУ, стр.129-135.

[Martin, 1975] J.Martin. Computer Data-Base Organization. Prentice-Hall, Inc., Englewood Cliffs, New Jersey / Дж. Мартин. Организация баз данных в вычислительных системах. Москва, Мир, 1978.

[Masser, 2005] Ian Masser. THE FUTURE OF SPATIAL DATA INFRASTRUCTURES. ISPRS Workshop on Service and Application of Spatial Data Infrastructure, XXXVI (4/W6), Oct.14-16, 2005 Hangzhou, China http://www.commission4.isprs.org/workshop_hangzhou/papers/7-16%20Ian%20Masser-A001.pdf

[Subieta, 2005] Kazimierz Subieta, Principles of modern database query and programming languages. (December 2005) http://www.sbql.pl/Topics/Principles%20of%20query%20programming%20lang.html

Authors' Information

Krassimir Markov – Institute of Mathematics and Informatics, BAS, Acad. G.Bonthev St., bl.8, Sofia-1113, Bulgaria; e-mail: markov@foibg.com

HOW TO USE A DESKTOP VERSION OF A DBMS FOR CLIENT-SERVER APPLICATIONS

Julian Vasilev

Abstract: DBMS (Data base management systems) still have a very high price for small and middle enterprises in Bulgaria. Desktop versions are free but they cannot function in multi-user environment. We will try to make an application server which will make a Desktop version of a DBMS open to many users. Thus, this approach will be appropriate for client-server applications. The author of the article gives a concise observation of the problem and a possible way of solution.

Keywords: Database management systems (DBMS), Information technology, parallel processing, Cache, client-server applications, application server, sockets.

ACM Classification Keywords: H.2.8 Database Applications, H.4 information systems applications.

Introduction

Single user versions of some DBMS are also called Desktop versions. They are usually free for commercial, home and office use. We can give Intersystem's Cache as an example [1]. The license fee for the use of the multi-user version of this DBMS is 245 EUR per process excluding VAT (Value Added Tax). If we calculate it with VAT then price is 294 EUR. If a company buys accounting software for 200 EUR and wants to use it as a client-server application for 4 computers, it has to pay 200 EUR for the software and 1176 EUR for license fees just for the right to use the multi-user version of the DBMS. This fact obstructs many small and middle enterprises in buying software products. That is why we have a possible solution. We will build an application server which will receive queries from workstations and redirect them to a single-user database. After receiving the answer from the database it will be redirected to the appropriate workstation. In this way, end users cannot feel that they use a single-user database. Moreover there is no need in changing the existing software, for instance the accounting software.

Background of the problem

Form a technological point of view this idea can be realized in Delphi, C#, Visual Basic. Moreover, there is a version of Delphi for Linux, called "Kylix". In this way the future application server can be compiled for another operating system. The program implementation consists of two parts: a server application and a client application. According to Cantu [2, 749] the idea can be realized by using the communication interface DCOM.

"DCOM is directly available in Windows NT/2000 and 98/Me, and it requires no additional run-time applications on the server. You still have to install it on Windows 95 machines. DCOM is basically an extension of COM technology that allows a client application to use server objects that exist and execute on a separate computer. The DCOM infrastructure allows you to use stateless COM objects, available in the COM+ and in the older MTS (Microsoft Transaction Server) architectures. Both COM+ and MTS provide features such as security, component management, and database transactions, and are available in Windows NT/2000 and in Windows 98/Me. Due to the complexity of DCOM configuration and of its problems in passing through firewalls, even Microsoft is abandoning DCOM in favour of SOAP-based solutions."

We made several experiments and few basic problems occurred. First, computers with different version of operating systems (for instance Windows 98 and Windows XP) cannot communicate. Second, DCOM does not keep alive several connections. Third, there is a limit in the number of connections. That is why our research continues. We want to find a better solution. Let us examine the work of a multi-user database (fig. 1).

The server part is usually installed on a computer, named "server" and the client part of the DBMS – on workstations. When we use a different DBMS the installation process is similar.

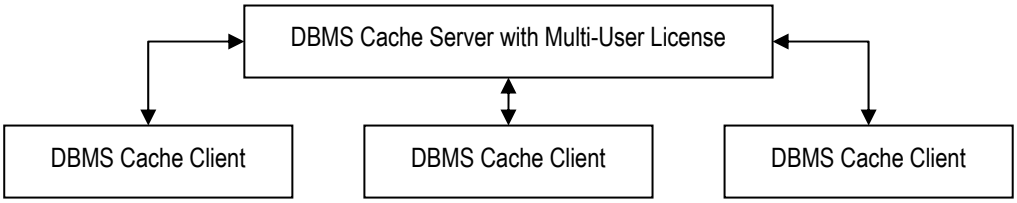


Figure 1: Technology of using a multi-user DBMS

A possible solution

We have to use another information technology to solve this problem. The communication can be realized using Transmission Control Protocol/Internet Protocol (TCP/IP for short). In a local area network (LAN) each computer has a unique IP Address. Connections between computers can be implemented by TCP ports. Each TCP connection takes place though a port. Some TCP ports have a standard usage for specific high-level protocols and services. In other words, you should use those port numbers when implementing those services and stay away from them in any other case. Here is a short list (table 1).

Table 1 Ports for some protocols

Protocol	Port
HTTP (Hypertext Transfer Protocol)	80
FTP (File Transfer Protocol)	21
SMTP (Simple Mail Transfer Protocol)	25
POP3 (Post Office Protocol, version 3)	110
Telnet	23

HTTP, SMTP and FTP are standard protocols. If we want a custom communisation between server and workstations we have to define custom protocol. A set of communication rules is generally indicated as a protocol. Basically, the server can receive different requests and, depending on the type of request and whether it can be accomplished, replies to the client. The server will respond to many requests. Transfer protocols are at a higher level than transmission protocols. That is why protocols are independent not only from the operating system and the hardware but also from the physical network. Communication can be started only if we launch a server program which accepts client connections. The client requests a connection indicating the server it wishes to connect to. When the client sends the request, the server can accept the connection, starting a specific server side socket, which connects to the client-side socket.

Methodology of implementation

Delphi 5 ships with three sets of socket components. Newer versions of Delphi also support Socket components. They can be used to read and write information over a TCP/IP connection. The Internet page of the palette hosts the Client Socket and Server Socket components. Sending text to server can be done by issuing method "SendText".

*ClientSocket.Socket.SendText('Select * From Customers Where CustNo = 1394')*

In this way we can send to the server a SQL (structured query language) statement. The server will receive the sent message as simple text. The Server Sockets reads the text by calling the method "Client Read". The text is actually contained in the property "Receive Text".

SQL_to_execute := ServerSocket.Socket.ReceiveText;

The server can use blocking or non-blocking connections. When the server uses blocking connections requests are processed in sequence. Huge information systems cannot scale by blocking connections. One of the possible solutions is the use of non-blocking connections. If we build a large system a good idea is to use threads to communicate with the database.

For the starting of the server we have to do the following:

ServerSocket.Port := 1974;

ServerSocket.Active := True;

On the client side, to connect to the server we have to connect to the server:

```
ClientSocket.Port := 1974;
ClientSocket.Host := '192.168.23.117'; // This is the IP address of the server
ClientSocket.Active := True;
```

As we mentioned we send a SQL statement to the server.

ClientSocket.Socket.SendText(SQL_to_execute);

The server receives request. This event fires the method OnClientRead of the Server Socket.

```
procedure TForm1.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
var
  i:integer;
  st : string;
begin
  for i := 0 to ServerSocket.Socket.ActiveConnections-1 do
  begin
    with ServerSocket..Socket.Connections[i] do
    begin
      st := ReceiveText;
      if st <> '' then
      begin
        Memo1.Lines.Add(RemoteAddress + ' sends :') ;
        Memo1.Lines.Add(str);
      end;
    end;
  end;
end;
```

In this way received text is added in a Memo.

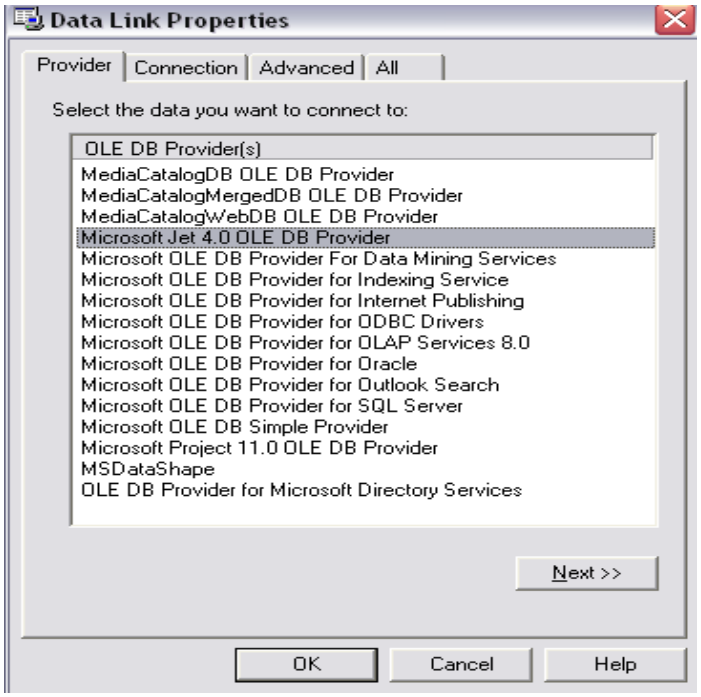


Figure 2 Provider for Data Link Properties

We have to redirect it to a database. We can use ADOConnection and a Windows UDL file to access different types of databases (for instance MS Access, Oracle, MS SQL Server, Informix, Sybase, Cache, DB2 and others).

To make an UDL file we just make a simple text file and we change its extension from "txt" to "udl". We open the file. On the "Provider" tab we set the type of DBMS we want to use (fig. 2).

The marked choice is for MS Access. If we use Oracle, we have to choose Microsoft OLE DB Provider for Oracle. If we use MS SQL Server, we have to choose Microsoft OLE DB Provider for MS SQL Server. In the next step we choose the database, username and password for login and we can test the connection. In this way our server application is DBMS independent. Moreover, the connection to the database is initialized through a text file which looks like an "ini" file. It is a stand-alone file and can be modified without the need of compilation.

As we highlighted the server receives requests. They are redirected to a DBMS using "ADOConnection" and "ADOQuery". The result of the execution of a query is in the form of DataSet. This dataset is two-dimensional. It consists of rows and columns. To be send back to the server it has to be represented as a simple string. That is why we have to use 2 delimiters: one - for rows and another one – for columns. They can be "tab character" – ASCII code "9" for field delimiter and "line feed and carriage return" – ASCII codes "10", followed by "13" – for record delimiter (table 2).

Table 2 Simple dataset returned as a result of execution a query

Order_number	Order_date	Cust_code
12345	03.04.2007	1395
12336	04.04.2007	1391

This tabular data will be transformed in one string as follows:

Result_string := '12345'+#9+'03.04.207'+#9+'1395'+#10#13+'12346'+#9+'06.04.207'+#9+'1391';

Actually, the result string is formed by using two "for" cycles. The sample code is too simple. That is why we skipped it. The next step is sending the result dataset to the client.

ServerSocket.Socket.Connections[nConnection].SendText(Result_string);

The variable "nConnection" indicates the unique number of connection. The client socket receives the result dataset. The Client Socket fires the event "OnRead". To read the incoming message from the server we have to write the following:

Received_text := socket.ReceiveText;

The next step is to convert the string into a two-dimensional array in order to visualize the dataset in a tabular format. This operation is simple. That is why we go on. A corner-stone can be the size of the string send over socket connection. A possible solution is to send the result dataset in several parts. OLE fields are another corner-stone. If we have to send a file or multimedia fields or BLOBs (binary large objects) we can use streams over socket connections.

Software development in brief

To realize the idea of using a desktop DBMS in a local area network (multi-user environment) we need to do the following. Firstly, we have to use the technology of communication by using the built-in port in Windows. For instance Trojan horses and some DBMS (Cache uses port 1972) use them for communication. Likewise, web servers use this port to communicate with end users. The default port is 80. This fact determines the use of sockets. We can use Client Sockets and a Server Socket. Secondly, we need an application server which will stay one level above the DBMS and will dispatch queries. Its kernel is based upon a Server Socket. Thirdly, a small application is needed on workstations with built-in Client Socket. Fourthly, we need a standard for communication between end-users' workstation and the application server. An example of such organization of work is described in fig. 3.

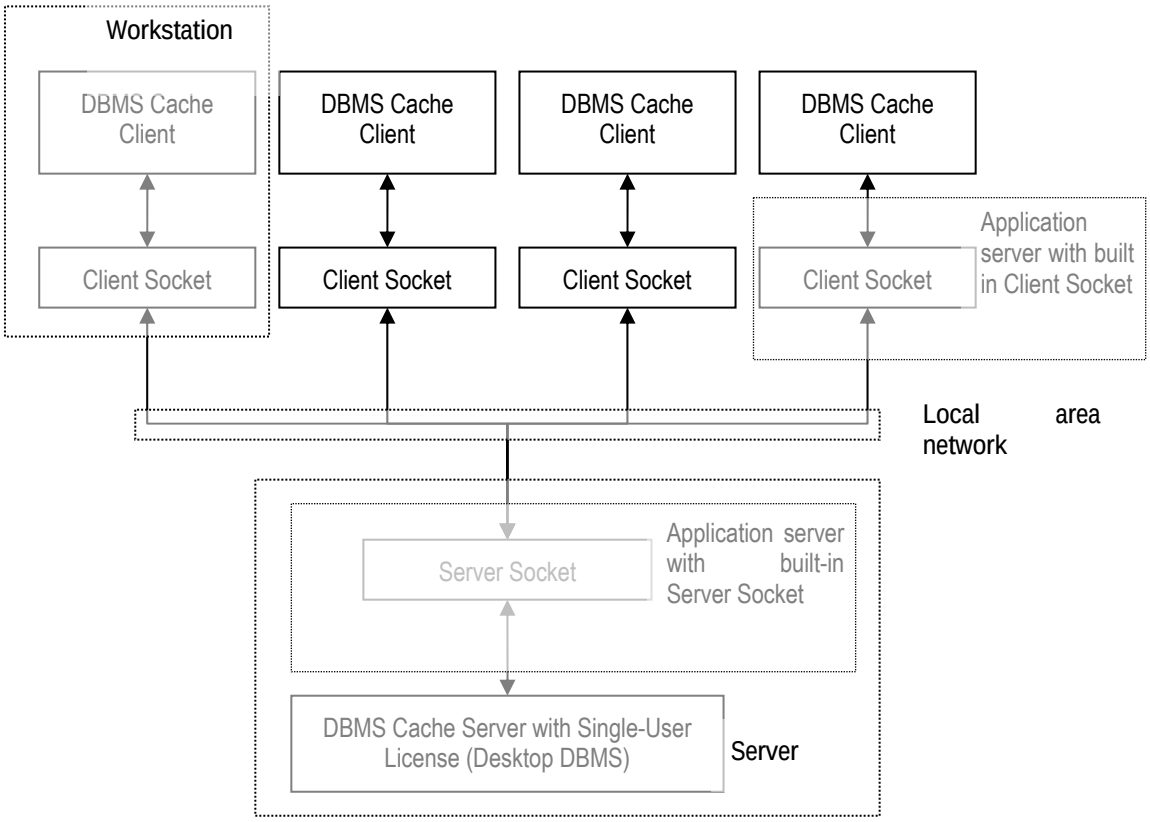


Figure 3: Technology of using a single-user DBMS in multi-user environment

In another aspect this application server can be used as a dispatching server for cluster applications. This server can have several subordinate sub-servers which process users' queries. In this way we get better parallel processing of queries. The result is similar to that in search engines such as google.com, yahoo.com, live.com, ask.co.uk.

Conclusions

As the reader sees, we succeeded in using a single user DBMS in multi-user environment. We gave a methodology of implementation of this idea. It was realized in a software product installed in 2005 at the Varna University of Economics – graphic user interface software for testing students' knowledge. The system is used in 7 disciplines. The achieved positive results are obvious evidence for the positive aspects of issued concepts. Moreover, software companies in Bulgaria can easily adapt this idea and make their products accessible to many small and middle enterprises.

Bibliography

[1] www.e-dbms.com.

[2] Cantu, M. Mastering Delphi 6, Sybex, Alameda, CA, 2001.

Authors' Information

Julian Vasilev – Chief assistant professor, Department of Informatics, Varna University of Economics; 77, Kniaz Boris I str.; Varna; Bulgaria; e-mail: vasilev@ue-varna.bg

TABLE OF CONTENTS OF VOLUME 2, NUMBER 1