I T H E A

International Journal INFORMATION THEORIES & APPLICATIONS

2005 Volume 12 Number 3

International Journal
INFORMATION THEORIES & APPLICATIONS
ISSN 1310-0513
Volume 12 / 2005, Number 3

IJ ITA is official publisher of the scientific papers of the members of the Association of Developers and Users of Intellectualized Systems (ADUIS).

IJ ITA welcomes scientific papers connected with any information theory or its application. Original and non-standard ideas will be published with preferences.

IJ ITA rules for preparing the manuscripts are compulsory.
*The **rules for the papers** for IJ ITA as well as the **subscription fees** are given on* www.foibg.com/ijita.
The camera-ready copy of the paper should be received by e-mail: foi@nlcv.net

Responsibility for papers published in IJ ITA belongs to authors.

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

International Journal "INFORMATION THEORIES & APPLICATIONS" Vol.12, Number 3, 2005
Printed in Bulgaria
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, BAS, Bulgaria.
Publisher: FOI-COMMERCE - Sofia, 1000, P.O.B. 775, Bulgaria. www.foibg.com, e-mail: foi@nlcv.net

ISSN 1310-0513

STATIC AND DYNAMIC INTEGRATED EXPERT SYSTEMS: STATE OF THE ART, PROBLEMS AND TRENDS

Galina Rybina, Victor Rybin

Abstract: *Systemized analysis of trends towards integration and hybridization in contemporary expert systems is conducted, and a particular class of applied expert systems, integrated expert systems, is considered. For this purpose, terminology, classification, and models, proposed by the author, are employed. As examples of integrated expert systems, Russian systems designed in this field and available to the majority of specialists are analyzed.*

Keywords: *integrated expert systems, real-time, simulation modeling, object-oriented model, rule, complex engineering systems, software tools, task-oriented methodology.*

ACM Classification Keywords*: I.2.1 Artificial Intelligence: Applications and Expert Systems*

Introduction

In the mid-1980s and early 1990s, the scientific and commercial success of expert systems (ES), one of the rapidly developing directions of artificial intelligence (AI), showed that, in actual practice, there is a sufficiently large class of problems that cannot be solved by the methods of conventional programming (for example, it is impossible to formulate the solution of the problem in mathematical terms in the form of a system of equations). This class is rather important, since the overwhelming majority of problems are very important in practice. In other words, if, 20 years ago, 95% of all problems solved by computer methods were problems that had an *algorithmic solution,* (hen, under modem conditions, especially in the so-called *describing application* domains (medicine, ecology, business, geology, design of unique equipment, etc.), the situation is completely different—*nonformalized problems* (NF-problems) make up a considerable amount of problems. These problems have one or several of the following characteristics: they cannot be given in a numerical form; the goals cannot be expressed in terms of a strictly defined goal function; and an algorithmic solution exists, but cannot be used due to the limited resources (time and/or memory).

However, NF-problems are not isolated from formalized problems (F-problems) as it is supposed in the basic concept of ESs (KNOWLEDGE + INFERENCE = EXPERT SYSTEM), but are a constituent of problems of real complexity and importance that are solved by conventional methods of programming. In practice, especially taking into account the constantly growing complexity of modern technological and management-technological systems and complexes, this results in combining in the framework of a 'joint programming system such diverse components as ESs and DBs comprising engineering, manufacturing, id management data; ESs and applied software packages (ASP) with developed computing, modeling, and graphical tools; ESs and learning systems including ring and learning systems; ESs and simulation sys-s for dynamical applications; ESs and developed hypertext systems; etc.

It is worth noting that *integration processes* that initiate important new classes of software systems are characteristic not only of ESs, but of all AI. They considerably expand the theoretical and technological tools are employed in designing *integrated intelligent stems* of various types and destinations, which was stressed in the papers of Pospelov, Popov, Erlikh, Tarasov, Kusiak and, later, Val'kman, Emel'yanov and Zafirov, Vittikh and Smirnov, Fominykh, Kolesnikov, Jackson, as well as other Russian and foreign specialists that have considerably contributed to the investigation and concept formulation of integrated intelligent systems.

It is worth noting that the trend towards integration of the AI paradigm with other scientific-technological paradigms when designing applications of intelligent systems has activated processes of so-called *hybridization* connected with designing *hybrid methods of knowledge representation*. The ideas of designing various hybrids and hybrid systems have been known for a fairly long time in AI. However, real practical applications have

only recently emerged. For example, the design of *neural expert systems,* that combine neural network methods and models for searching for solutions with mechanisms of ESs based on expert (logical-linguistic) models of knowledge representation and models of human reasoning became possible due to modern platforms and tools like Neur On-Line (Gensym Corp). However, despite successful individual theoretical investigations and implementations, there is no adequate theory and technique *of integration* and *hybridization* in AI, which partially explains the terminological confusion in using such terms as *integrated system* and *hybrid system* [Rybina,2002].

Thus, the trend towards integration of investigations in different fields in recent years has led to the necessity of combining objects, models, concepts, and techniques that are semantically different. This has inevitably generated both completely new classes of problems and new *architectures of software systems* that cannot be implemented by a straightforward usage of the methodology of simple ESs and tools like the *shell* of ESs that supports this methodology, which is inefficient, labor consuming, and expensive [Rybina,2004].

However, despite the obvious advantages of the design of IESs that can solve real practical problems in the framework of a unified architecture of a software system ensuring efficient data processing based on the interaction of logical-linguistic, mathematical, informational, simulation models, etc., this problem has not left the stage of problem statement.

Isolated attempts to solve the mentioned problems for IESs have been made both abroad and in Russia. However, most of the investigations and developments encompassed only part of these problems and cannot' pretend to propose a particular methodology for designing IESs. Among the Russian fundamental investigations in the field of producing elements of the IES concept, the papers of Erlikh and, from the standpoint of the methodology and technology of designing the software of conventional ES, the results of Khoroshevskii are the best known. Among the foreign works, we note the G2 tool system (Gensym Corp.), which can be used for designing dynamical IES. However, this system does not support a number of important tasks connected with computer knowledge acquisition and automatic generation of a knowledge base (KB). It also does not deal with the problems of designing applied IESs for various classes of tasks typically used in the IES concept.

For the authors of this paper, static IESs became an object of research and development in the late 1980s [Rybina,1997]. Later, in the mid-1990s, I started my research in the field of real-time dynamical IESs (RT IESs) [Rybina,Rybin,1999]. During this period, based on the experience of designing a number of IESs (from simple static IESs and others to dynamical RT IESs for such important and resource-consuming tasks as diagnostics of technological objects, management of industrial complexes and control of technological processes, ecological monitoring, etc.), a scientific direction that has its own object of investigation, conceptual, theoretical, and methodological basis has been formed. A set of methods and software tools (the AT-TECHNOLOGY system) that enables us to support the complete life cycle of automatic designing IESs for particular classes of tasks of solved problems and types of application domains on the basis of a particular task-oriented methodology of designing IESs in static and dynamical application domains has been developed [Rybina,2004].

Characterizing the proposed methodology in short [Rybina,1997], it is worth noting that it is based on a multilevel model of analysis of the integration process in IESs, modeling of particular types of tasks, the relevant techniques of conventional ESs (a task-oriented approach), and the methods for designing a model of the architecture of IESs and the corresponding software components at each integration level. This experience allows one to analyze rather systematically and visually the basic types of integrated intelligent systems (that comprise components similar to simple ESs) irrespective of the specific features of the particular application domain employing only a principle conditionally called "from the standpoint of the architecture."

However, the goal of this paper is not to attempt to make an *inventory of* all types of existing systems of the IES class. The survey of functional, structural, and, partially, programming specific features of various architectures of IESs (using examples of Russian systems) proposed below, in addition to presenting the state of the art for the field, is mainly directed towards finding ways for overcoming the technological lag in designing ESs described above as compared with modern technologies of designing and programming informational systems. This survey may also be useful in order to help to solve the problems of mutual semantic adaptation of scientific terms and notions that can be used by specialists in different fields in the development of applications on the basis of ESs.

1. Classification of IESs, the Interrelation of the Integration and Hybridization Processes in IESs

At present, we can observe two main processes in AI. One of them is connected with the expansion of architectures of intelligent systems by supplementing them with new tools from other scientific fields (the *integration process).* The other process is associated with the initiation of new subfields from known fields and the development of special new tools for them (the division process). No doubt, *integration processes* are of the most practical importance, since they join different approaches and paradigms, and, as a consequence, totally new classes of software systems.

An IES is a software system whose architecture, together with a conventional component, an ES that, as a rule, uses the ideology of simple production ESs in order to solve NF-problems, comprises certain components "N" that *extend functional capabilities* of ESs (e.g., a database, packages of application programs, learning systems). Therefore, all IESs can be split into two subclasses: IESs with *shallow component integration* and IESs with *deep component integration*.

Then, in the case of shallow integration of the components of ES and N components, the interaction between, e.g., the DBMS and ES can be carried out at the level of sending messages, and the results of the operation of each component are the source data for the other. In a more complex variant of the shallow integration, one of the software components (ES or N) calls the other in order to refine data or to solve some tasks. In this case, a closer interface is implemented, e.g., with the help of special programs that provide a bridge between the DBMS and ES for messaging. Examples of IESs with shallow integration are presented in Section 2.

The deep integration of ES and N components means that these components are improved by functions that are not characteristic of the conventional types of these components. For example, if we solve the integration problem improving the shell of an ES by including certain functions of a DBMS, this may mean that we set a goal to incorporate a generalized DBMS inside a knowledge base (KB). This approach is the most interesting if the KB is too large to be stored in the computer memory and when efficient mechanisms of data access are required. On the other hand, in relation to an improvement of the DBMS by incorporating basic functions of ESs, we should keep in mind that the DBMS has a developed specialized mechanism for executing such tasks as data storage, maintenance of data integrity, knowledge base navigation, and efficient servicing of complex queries. In addition, this mechanism can also be supplemented by functions of the ES. In particular, one of the approaches to solving this problem is to extend the DBMS in order to transform it into a generalized inference mechanism. It is worth noting that, for designing an IES with a deep integration of components, the approach associated with improvement of tools for ESs by including unconventional functions seems to be more preferable, which, in particular, was employed in the development of the task-oriented methodology for designing IESs.

Complete integration is the highest level of integration for IESs and joins the best properties of the ES and N components. For example, the complete integration of a DBMS and an ES consists in selecting the best specific features and mechanisms for designing the DMBS and the advantages of the ES and in designing new data and knowledge models in order to develop completely new systems with new capabilities. The undoubted advantage of this approach to integrating DBMS and ES components is that all components belong to one system with a unified structure for modeling facts and rules and uniform data and knowledge processing.

The so-called *multilevel model of integration processes* proposed in the framework of the task-oriented methodology allows one to analyze the trends towards integration and hybridization in modern IESs. This methodology considers the integration processes from the standpoint of the following key aspects:

- integration in the architecture of IESs of different components that implement both F-problems (the N component) and NF-problems {the ES component) and determine the specific features of functioning of the IES *(high level of integration);*
- integration *(functional, structural, and conceptual)* connected with the concepts and methodologies of designing and developing particular classes of IESs and their components *(middle level of integration);*
- integration (informational, programming, and technical) connected with the technologies, tools, and platforms (low level of integration).

For at least three levels of integration in IESs, this model allows us to trace the interrelation of particular methods of integration of some components of the IES (N or ES) with the necessity of internal integration,

i.e., *hybridization* of some models, methods, procedures, algorithms, etc. In this connection, the following conclusions and suggestions have been made:

- the notion of integrated intelligent systems differs from the notion of hybrid intelligent systems;
- an integrated intelligent system must not be hybrid and vice versa;
- an integrated intelligent system is always hybrid only in the case of complete integration of components.

Therefore, the term "integrated expert system" should be used for complex programming systems that join methods and facilities of ESs with the techniques of conventional programming. The term "hybrid expert system" is advisable to use both for ESs and tools for ESs with hybrid methods of knowledge representation (i.e., joining inside one tool different models of knowledge representation and mechanisms of functioning).

These recommendations are apropos and useful, since the lack of semantic unification leads to misunderstandings in the usage of terms by specialists. Some problems in using the terms "hybrid" and "integrated" may be explained by historic reasons connected with the early paper, where IESs were defined as "hybrid ESs."

It is worth noting that the consideration of problems of shallow and deep integration of the base component ES with a component N (where N is a DBMS, ASP, etc.) at the *high* level allows one to construct in general the so-called generalized model of the architecture of the designed system that reflects its structure. The *intermediate level* is of the greatest interest, since, in the framework concepts and methodologies of IESs, it provides an opportunity to consider the distribution of functions among the ES and N components when developing an application, as well as some design decisions over all components of the designed system. In particular, we can consider possible variants of the joint functioning of the ES and N components. There are several possibilities: the ES and N can be included in each other; they can be used in parallel to solve the same problem; they can be partially included in each other; and an intelligent interface between them can be organized. The integration at the *lower level* is provided by the technology and capabilities of the tools used in the course of development.

Let us explain the terms "static IES" and "dynamic IES" beginning from the notions "data domain" and "application domain." It is well known that "data domain" means a domain of human activity that is specially selected and described (i.e., a set of entities that describe the domain of expertise). The notion *application domain* includes the *data domain* plus a totality of problems solved in it. If an IES is based on the assumption that the source data about the data domain that provides a background for solving the particular problem does not change in the course of solving the problem or the problems solved by the IES do not directly change knowledge about the data domain in the course of their solution, then this domain is *static* (i.e., the domain has a static representation in the IES), while the tasks solved by the IES are static. In other words, a *static IES* is supposed to operate in a static *application domain* if a static representation is employed and static tasks are solved. Otherwise, we deal with a *dynamic IES* that operates in a *dynamic application domain.* This means that a dynamic representation is used and dynamic problems are solved.

Below, based on the introduced terms, notions, and classifications, examples and specific features (functional, structural, and, partially, programming) of the design of a number of Russian and foreign IESs are considered corresponding to the types of N components.

2. Analysis of the Specific Features of the Design of an IES

The analysis of foreign and Russian developments of IESs has shown that almost all of them are devoted to the description of applications of IESs that combine in their structure a set of components that implement both F-tasks and NF-tasks; i.e., they are systems with a *high integration level* (an integration within the scope of the general architecture of the IES). Below, we consider some examples.

2.1. Integration of an ES and a DB

It is worth noting that, despite the fact that, historically, ESs in artificial intelligence and DBs have been developed separately, nevertheless, there are many application domains where, simultaneously, both access to an industrial DB and the use of an ES for decision making on the basis of experience or expertise are required. In this connection, the problems of integrating *static* ESs and DBs have been investigated the best. Therefore, at present, there are many publications devoted to a specialized application of IESs that combine the techniques of KBs and DBs. However, most authors have concentrated their efforts on the description of ways of implementing peculiarities of one or another application domain and have paid substantially less attention to the principles,

methods, and models of integration in the framework of ESs and DBs. In this connection, we place the main emphasis only on the papers that are of interest from the standpoint of methods of joining ESs and DBs (DBMSs).

There are two approaches to joining ESs and DBs—*weak and strong coupling*. Strong coupling is used when there is an opportunity to split the operation of an ES into a sequence of steps that require definite predictable and limited information from the DB (e.g., in tasks of medical diagnosis, the information about a particular patient can be requested before the beginning of a consultation). As a result, the opportunity to efficiently arrange the operation of the ES is a merit of this approach, since, in the course of consultations of the ES, there is no need to query the DB.

However, in other application domains, this approach may be impossible and a strong coupling of the ES and DB may be required. This coupling allows one to multiply read and modify the content of the DB. It is quite obvious that a strong coupling imposes more serious requirements on the system; in particular, the rate of transactions between the ES and DB supposes an efficient implementation of this functional component. Among the first Russian systems of this class, the shell for the DI*GEN ES that has a mechanism of strong coupling of the ES and DB is worth mentioning. To implement this mechanism, DI*GEN contains a set of functional primitives that allow one to position the DB, to read and modify data from an arbitrary field of any record of the DB, to add/delete any records in the DB, retrieve data using a condition, etc.

A radically different approach to integrating ESs and DBs was implemented in one of the earliest Russian applied ESs, the integrated intelligent system of the molecular design of physiologically active substances, which was also *hybrid,* since it employed several reasoning strategies and models of knowledge representation. From the standpoint of the technique for joining an ES and a DB, the characteristic specific feature of this system is the use of an original relational DBMS for generating training samples and storing data, since the decision strategy in this IES is based on two complementary concepts of inductive learning by examples.

Among other developments in the field of static IESs of the early period, a large group was presented by the systems, in which the integration between the ES and DB, ES and ASP, and ES and other components was obtained by using integrated tools like the well-known GURU (INTER-EXPERT) system that contains facilities for designing static ESs, built-in DBMSs, electronic worksheets, and graphical packages. In GURU, the *premise* of each rule in the KB can contain working variables with one value, multivalued fuzzy variables, static variables, cells of electronic worksheets, fields of the KM AN relational DB, numerical functions, operators of relations, Boolean operators, operators over numbers, operators over strings, and symbols, thus providing the integration. Thus, using GURU, from the standpoint of the *lower level of integration,* numerous components N can be joined with components of the ES in the scope of one operation, which allows one to efficiently arrange processing of various data in applied IESs both together with the ES and irrespective of it. This also eliminates the necessity to write special bridges for the IES.

In later *dynamical* IESs that operate in real time (RT IESs), the integration of the ES and DB at the high level is based on designing special bridges, e.g., when using G2 (Gensym Corp.), the G2 Database Bridge is used. For example, when designing subsystems for data acquisition and storage for a system of on-line monitoring, the capabilities of integrating G2 and DBMS Oracle through the G2-Oracle (Oracle Bridge) interface were used. In addition, the tools of the DBMS Oracle and the system software support data backup and long-term storage. In G2, the transmission of online data from external sources was supported by the standard interface of G2 (GSI).

2.2. Integration of ESs with Learning Systems

The next large group is represented by IESs whose architecture integrates ESs with various kinds of learning systems. In general, in connection with the problem of designing computer learning systems, which emerged earlier than ESs and have gone a long way from laboratory programs to power commercial systems, we cannot avoid mentioning the following important specific feature. There are two main processes of learning—"learning" and "tutoring"—that substantially affect the approaches of the soft implementation of IESs.

The direction connected with "learning" process *(learning systems)* includes self-learning, learning with a teacher, adaptation, self-organization, etc. Therefore, when designing learning systems, sufficiently strict models (perhaps, hybrid) are investigated and constructed. These models show the ability to adapt to the external world by storing information. However, from the standpoint of this survey, the direction connected with "tutoring" process *(tutoring systems)* is the most interesting. In this direction, models of the transmission of information from a teacher with

the help of a computer are investigated. This is especially interesting, since, in pedagogic, there is no commonly accepted learning theory or formal models of a trained person, tutoring, teaching effects, explanations, etc. Due lo these facts, specialists depend on expert models and tools of their implementation in the scope of IESs.

Tutoring ESs and expert-tutoring systems are the most popular, as well as the entire field called "intelligent tutoring systems" (ITS), among others. It is worth noting that the interpenetrating of integration processes in AI and pedagogic was reflected in ITSs, as well as in tutoring ESs (which can be considered as a subclass of ITSs). This interpenetrating stresses the need for additional components N that can support a *model of a trained person.* Based on this model, the teacher specifies a current subgoal at the strategic level and the components that realize a particular *tutoring model* in the form of *tutoring actions* at the tactical level. These components must also provide an opportunity to the teacher to observe the actions of the trained person and to help him by using the developed interface facilities.

However, none of the existing systems is able to completely implement the ideas of deep and, especially, complete integration of ESs and N components that support the specified models. In the best case, successive functioning of autonomous components that implement the simplest model of the trained person and a number of learning actions (one of which was "training" based on an ES) was provided.

2.3. Integration of ESs with Hypertext Systems

A wide range of papers is connected with the description of various IESs whose architecture includes integration of components of ESs with not only DBs and tutoring components, but components with hypertext facilities (HT-facilities) as well. The appearance of hypermedia and multimedia systems and interactive leaching environments in recent years has led to the necessity of developing a new type of integrated system—*intelligent teaching environments*—that combine the capabilities of ITSs, hypermedia systems, and teaching environments.

Modern investigations devoted to the problems of integrating ESs with HT-systems provide another sufficiently large range of investigations. Their practical orientation becomes more obvious under the mass distribution of Internet technologies, electronic encyclopedias, Internet shops, etc. The deep integration of the components of ISs and HT-tools actually supplement the capabilities of ESs for hypertexting and allow one to make logical inferences in searching for relevant text fragments, especially, in the fields of activity recommended by standard documents and in electronic business. For future applications, IESs of this type can evolve to "intelligent texts" or "expert-texts," and adaptive models of a user can be designed.

2.4. Integration of an ES and an ASP

Let us consider the analysis of processes of integrating an ES and an ASP in an IES. The global problem of designing *intelligent computer-aided design systems* (ICAD) has been of interest for a long time for researchers and designers. The number of publications devoted to this topic is very large. It is the scope of these papers where the problems of deep integration of ESs with developed mathematical, computational, modeling, and graphical tools, implemented, as a rule, in the form of ASP, were first posed. In such complex software systems, a conventional ES should play the role of an *intelligent user interface,* while the decision maker of the ES should service not only the KB, but also be a *monitor* for the ASP, which can be supplemented not only from the external world, but as a consequence of the operation of the ES.

However, despite the large number of papers that describe applied IESs for various but rather narrow applications, the problems of integrating ESs and ASPs are mainly solved in the scope of IESs with *shallow integration,* i.e., at the level of message exchange between components. The PRIIUS system that joins in its architecture an ES, a data retrieval system, and programs in C, Pascal, and FORTRAN based on the principles of complicated shallow integration is a typical example.

A large number of Russian IESs of this type use foreign ASPs and shells of ESs. For example, in surveys, a large group of IESs applicable in problems of analysis, design, and optimization of the structure of industrial structures and objects of building was described in detail. Among these systems are SACON, designed based on a shallow integration of the well known shell EMYCIN and the MARC package of structural analysis; HI-RISE, in which an ES functions in the DICE environment of the graphical interface; PROTEI, in which the ES plays the role of a monitor that provides the organization of the interaction of all subsystems and components of the IES; and BTEXPERT, in which an interactive FORTRAN program is combined with a ES shell written in Pascal/VS and the processing of symbolic data and the interface are implemented by a pair of mutually complementing tools produced by IBM (ESCE and ESDE). For other classes of problems (control problems), such systems as

CACE-III implemented with the help of DELPHI and having access to an ASP of numerical analysis (CLADP, SSDP, SIMMON, and SEPACK) and CASCADE implemented with the help of the DECIDE shell and working with an ASP library that comprises 78 programs of numerical analysis written in FORTRAN are well-known developments.

However, as was shown in this paper, to describe the set of *typical situations* for each type of aircraft, a totality of mathematical models is used in the OOAES. To support these models, an autonomous functional unit, connected with a rule base and a mechanism for making decisions in a rather simple way (like that of GURU) by including in the working memory (data base) of the OOAES elements of the corresponding mathematical models and using them in the left-hand sides of rules-productions, was developed. In (his case, the valuated elements of the working memory correspond to the actual values of the output signals from the unit for supporting models; the mechanism of logical inference in the OOAES—processing rules—initiates the execution of autonomous units of the support of mathematical models and, then, continues the search for a solution in the rule base. Thus, mathematical models are called from the decision-maker at the level of external programs. In this connection, early OOAESs can be referred to as IESs with *complicated shallow integration* of components of ESs and ASPs (in this case, by mathematical models and the facilities for their support) with further perspectives for deep integration of components and, perhaps, hybridization on the basis of hybrid models of knowledge representation and reasoning.

It is worth noting that the ideas of *integration and hybridization of diverse models* in constructing applied intelligent systems (including IESs, stipulated by modeling real objects, setting and solving problems on certain models, integration of models, problems, and solutions obtained) were initially in the background of AI, which has been mentioned several times. The design of a unified informational environment and formulation of nine features of integration allow one to analyze the interaction of a user, complex product, and data resources and processes from the standpoint of design, theoretical, and experimental investigations on the basis of synthesis and analysis of models (and their junction). Nevertheless, despite the general principles proposed and particular examples of implementation, the problem of designing IESs with complete integration of components, i.e., hybrid systems, remains the most complex problem.

2.5. Integration of ESs with Simulation Systems

The developments that join conventional ESs and simulation systems (SSs) in the framework of *dynamic* IESs are the most widespread. This is the case in which the necessity and possibility of integrating methods for solving NF- and F-problems reveal themselves in the most complete way. In particular, these two techniques are considered, their similarities and differences are analyzed, the expediency of their mutual complementing and interaction is substantiated, and classifications of possible approaches and integration of the ES and N components are proposed from the standpoint of the *intermediate level of integration* (built-in systems, parallel systems, cooperating systems, and intelligent interfaces). A dynamic real time IES that comprises N components represented by a subsystem for modeling the external world is a typical example of an IES designed on the principles of integrating ESs and SSs.

It should be noted that dynamic IESs that are used to support solutions of NF-problems and AI problems are the most complex systems from the standpoint of implementation. As a rule, either *complicated shallow integration* of certain components of the IES or *deep integration* is used. The problems of complete integration are partially discussed only at the conceptual level.

The facilities of simulation are incorporated not only in the architecture of IESs. Very frequently, simulation systems are included in CAD systems, since, due to the complexity of the designed system, only simulation methods allow one to use relevant information of various kinds, including exact data and quantitative data, as well as expert, heuristic knowledge-based experience and assessments. In this case, the structure of the CAD system, together with SSs, contains a certain number of ESs and a common DB.

2.6. Integration of ESs with Knowledge Acquisition Systems

The advances in the ES field have led to the growing importance of special methods and software tools for knowledge acquisition (eliciting, extracting), especially in designing large knowledge bases. In other words, the direction of knowledge acquisition, which is conventional and key for AI, is separated out into a particular research field and attracts methods and approaches from other scientific fields such as psychology, linguistics, neural networks, machine learning, mathematical statistics, regression analysis, data visualization, etc.

An analysis of papers devoted to new lines of AI investigation shows that, beginning in 1994, many specialists specify software tools for knowledge acquisition into a separate category, directly connect it with data mining that rapidly develops knowledge acquisition in DBs, and predict great prospects for this field. In some cases, the systems that combine the capabilities of conventional ESs with a component for knowledge acquisition (and, sometimes, even with a DB) are proposed to be called "partner systems". However, there is no information about the methods and ways of integrating the ES and N components in the architectures similar to IESs in that paper.

2.7. Integration of ESs with Other Software Components

Consider other examples of integrating the ES and N components in the architecture of static and dynamic IESs that are not so characteristic. Under modem conditions, the integration processes manifest themselves most visibly in the architecture of systems aimed at reengineering business processes of enterprises. As is known, the main task of business process reengineering is to find a new way of constructing the existing business on the basis of employing the most modern data technologies, including the methods and tools of ESs. Here, it is important to take note of two aspects.

The first is associated with the general methodology of designing ESs, OOA, and business process reengineering. All of these methodologies are based on the same principles and the same life cycle and the models of processes of a new business that are designed in the course of reengineering correspond to analogous models that are constructed when designing software systems for supporting business and can serve as models of application domains developed at the stages of "identification" and "conceptualization" when designing ESs.

The other aspect is the close relation of the conventional paradigm of ESs of the paradigm oriented to the rules with the so-called business rules. The languages for knowledge representation of production types by and large have the required power for the representation of business rules, in contrast to, e.g., the tools of diagram techniques of structural analysis or object data models of OOA. We can describe all language constructions proposed, in particular, by Ross for describing business rules using the facilities of the language for knowledge representation used in the famous ReThink system. However, only preliminary studies have been undertaken in this area.

Conventional decision support systems (CDSS) have great prospects in the field of integrating the ES and N components, as well as in the hybridization of models of knowledge representation and methods of reasoning. This is especially true in connection with their evolution toward designing *intelligent* CDSS, which, as was mentioned by the authors, can be fully implemented only if the modem techniques for designing intelligent systems based on the concepts of distributed AI and dynamic knowledge models and the methods of plausible reasoning are harnessed. It is also necessary to employ powerful computational platforms and corresponding tool systems similar to G2.

Completing the survey, we should note that, in real practice, together with the considered cases, it is possible to find other examples of the use of models, methods, and tools of ESs in combination with conventional approaches and programs, e.g., integration of ESs with geoinformation systems oriented toward processing cartographic data. And, no doubt, the most modern agent-oriented paradigms in AI are based on the further evolution of intersystem integration processes.

2.8. New Architectures of Distributed IESs

From the standpoint of further prospects in the field of development and evolution of IESs, we should notice (he rapid progress in the field of Web-oriented IESs (Web-IESs), which, on the whole, fits the modem trends in designing intelligent systems connected with the transition from isolated autonomous systems with centralized management to distributed integrated systems with network control structures.

Up to now, the methods of interchange between a Web browser and a network has considerably complicated the design of Web-IESs, in particular, the functioning of the inference engine of an ES. However, currently, the technologies that allow one to solve these problems have begun emerging. At present, there are two main types of architectures of Web-IESs, whose specific feature is completely determined by the methods of activating the inference engine of the ES in the network. The first is the "server-master" architecture that stipulates the ES to be run on the server. The second architecture—"client-master"—stipulates the ES to be run on the client computer (local user computer). Let us evaluate the merits and demerits of each of these methods from the standpoint of designing the architecture of a distributed IES.

Despite the certain access rate connected with the availability of all programs required for operating an ES on the server (which allows us to eliminate the loading of (his program on the user computer), the server-master

approach implies a number of inconveniences. First of all, this is associated with the impossibility of meeting a number of conditions of normal functioning of an ES such as the impossibility of saving the history of the system's states and of tracing the steps of inferences made. Moreover, when several clients simultaneously work with the ES, the load of the server and response time to the user queries increase. When using the client-master method, data are processed on the user computer and thus eliminate the problem of storing the states and trajectories of the inference in the ES.

Consequently, each of these approaches has it advantages and disadvantages, and the future will show which variant will be preferable. It is clear that the server approach requires considerable resources on the server, while the design of Java-client ESs shifts all processing problems to the user computer. In addition, in order to prefer some of the considered variants of designing a Web-IES, it is necessary to know the amount and rate of data transmitted by the corresponding N component.

ES and N components are combined at the level of shallow integration by special bridges that can be implemented by standard techniques (COM, DCOM and CORBA), which makes the processes of interaction of an ES with other N components transparent and simple to implement.

At present, a number of foreign Web-IESs an known that are implemented by either the tools of conventional programming such as C++, Visual Basic, Active Server Pages, and Active X or using special tools like KROL (an object language for knowledge representation) or others.

Conclusion

The questions of to what extent modern expert systems and, in a broader sense, *knowledge-based system* are expert and what are the future trends and prospect: of the development of this class of intelligent systems the most popular from the beginning of the middle of the 1980s are of interest for researchers and designer; of ESs, as well as potential users. The euphoria from the first successes in designing conventional ESs for limited application domains was marred by the difficulties and technological problems when trying to solve problems of real practical complexity and importance.

This survey was devoted to one of the most important problems connected with the change and complication of the architecture of modern ESs because of the dominating processes of integration and hybridization (the emergence of integrated, hybrid, and Web-ESs). Naturally, tile limited length of this survey cannot give a complete representation of the variety of types of the existing architectures of IESs and approaches to their implementation, since it is the ES field where the greatest experience has been accumulated. This experience has become an essential part of almost all *integrated intelligent systems.* Note that we were primarily in the analysis of Russian developments in this area using sources available to any specialist that might be interested in the theory and techniques of the entire process of designing intelligent systems.

Acknowledgments

This work was supported by the Russian Foundation for Basic Research, project no. 03-01-00924.

Bibliography

[Rybina,1997] G.V.Rybina. Task-Oriented Methodology for Automated Design of Integrated Expert Systems for Static Application Domains, Izv. Ross. Akad. Nauk, Tear. Sist. Upr., 1997, no. 5.

[Rybina,2002] G.V.Rybina. Integrated Expert Systems: State of the Art, Problems, and Trends, Izv.Ross.Akad.Nauk, Teor.Sist.Upr., 2002, no. 5.

[Rybina,2004] G.V.Rybina. The new generation of software tools for application intellectual systems development, Aviakosmicheskoe Priborostroenie, 2004, no. 10.

[Rybina,Rybin,1999] G.V.Rybina, V.M. Rybin. Real-Time DynamicJB Expert Systems: The Analysis of the Research Experience and Developments, Prib. Sist. Upr, 1999. no. 8.

Author's Information

Galina Rybina – Moscow Engineering Physics Institute (State University),Kashirskoe shosse, 31, 115409, Moscow, Russia,Email: galina@ailab.mephi.ru

Victor Rybin – Moscow Engineering Physics Institute (State University),Kashirskoe shosse, 31, 115409, Moscow, Russia,Email: rybin@aie.mephi.ru

APPLICATION OF ARTIFICIAL INTELLIGENCE METHODS TO COMPUTER DESIGN OF INORGANIC COMPOUNDS

Nadezhda Kiselyova

***Abstract**: In this paper the main problems for computer design of materials, which would have predefined properties, with the use of artificial intelligence methods are presented. The DB on inorganic compound properties and the system of DBs on materials for electronics with completely assessed information: phase diagram DB of material systems with semiconducting phases and DB on acousto-optical, electro-optical, and nonlinear optical properties are considered. These DBs are a source of information for data analysis. Using the DBs and artificial intelligence methods we have predicted thousands of new compounds in ternary, quaternary and more complicated chemical systems and estimated some of their properties (crystal structure type, melting point, homogeneity region etc.). The comparison of our predictions with experimental data, obtained later, showed that the average reliability of predicted inorganic compounds exceeds 80%. The perspectives of computational material design with the use of artificial intelligence methods are considered.*

***Keywords:** artificial intelligence, computer design of materials, databases on properties of inorganic materials, information-analytical system.*

***ACM Classification Keywords**: I.2.1 Artificial Intelligence: Applications in Chemistry*

Introduction

Now the search for new inorganic materials is carried out, for the most part, on the basis of the experience and intuition of researchers. The problem of *a priori* prediction of compounds that have not yet been synthesized and evaluations of their properties is one of the most difficult problems of modern inorganic chemistry and materials science. Here the term "*a priori* prediction" means predicting yet unknown substances with predefined properties from only the properties of constituent components - chemical elements or more simple compounds.

The problem of predicting new compounds can be reduced to the analysis of the multidimensional array of the property values and the column vector of the desired property. Each row corresponds to some known chemical system, whose class is indicated by the row position of the column vector. For example, the multidimensional array can include the properties of chemical elements for set of known chemical systems with formation or without formation of compound with predefined composition (desired property). The process of analyzing all this information allows finding the classifying regularity. By substituting the values of the properties of the elements for unknown chemical system into the regularity thus found, it is possible to determine the class (compound formation or non-formation). So, the problem of *a priori* prediction of new compounds can be reduced to the classical task of computer learning. The chemical foundations of material computer design are based on Mendeleev's law which asserts that the periodic nature of changes in the properties of chemical systems depends on the nature and properties of the elements which makes these systems (compounds, solutions, etc). Another premise justifying the proposed approach is the existence of good classification schemes for inorganic substances.

Databases on Properties of Inorganic Materials and Substances as a Foundations of Material Computer Design

The application of computer learning methods for finding regularities is rather put into use in case of complete and qualitative initial data. Our experience of computer learning applications to chemistry shows that the number of erroneous predictions varies proportionally with ratio of number of errors in experimental data to number of learning set to be processed and the reliability of prediction grows with an increase of initial data volume (reliability mounts to a limit with an increase of size and representatives of learning set). Consequently, the application of the computer learning methods to chemistry implies the use of databases (DBs), containing

extensive bulks of qualitative information, as a basis. With this aim in mind, we develop the DBs containing data with the qualified expert assessment. The most interesting of them are DBs on materials for electronics with completely assessed information [Kiselyova *et al.*, 2004] and an inorganic compound properties DB containing partially assessed information [Kiselyova, 2002; Kiseleva *et al.*, 1996]. These DBs are integrated. The all our DBs have Internet-access (http://www.imet-db.ru).

1. A phase diagram DB of material systems with semiconducting phases "Diagram" [Kiselyova *et al.*, 2004] contains information on physical and chemical properties of the intermediate phases and the most important Pressure-Temperature-Concentration phase diagrams of semiconducting systems evaluated by qualified experts. Now the DB contains detailed information on several tens of semiconducting systems.

2). DB on acousto-, electro-, and nonlinear optical properties "Crystal" [Kiselyova *et al.*, 2004] contains detailed information regarding substances of these types evaluated by experts. In addition, DB includes extensive graphical information about properties of the materials.

3). A DB on inorganic compound properties "Phases" [Kiselyova, 2002; Kiseleva *et al.*, 1996] contains information about thermo-chemical and crystal chemical properties on more than 41,000 ternary compounds taken from more than 13,000 publications. Some of the data have been assessed by materials experts. This DB is a main data source for material computer design.

On the one hand, the use of DBs increases the reliability of predicting inorganic substances, and, on the other hand, the application of artificial intelligence methods extends the capabilities of databases owing to the search for regularities in the known information and the use of these regularities for prediction of new substances not yet synthesized.

Artificial Intelligence Methods for Material Computer Design

The search for, and development of effective material computer design systems were aimed at the creation of more powerful programs capable of analyzing, on the one hand, very large arrays of experimental information, and, on the other hand, of allowing construction of multidimensional classified regularities under the conditions of small sets. Improvements in electronics allowed the development of systems for chemical applications with a user-friendly interface, working in real time, for example, [Gladun, 1995; Gladun *et al.,* 1995; Chen *et al.,* 1999; Chen *et al.,* 2002; Pao *et al.,* 1999]. The trend has been a transition from the simplest algorithms of pattern recognition [Gulyev *et al.,* 1973; Kutolin *et al.,* 1978; Savitskii *et al.,* 1968; Talanov *et al.,* 1981; Vozdvizhenskii *et al.*, 1973] toward more powerful methods based on the use of neural and semantic networks [Kiselyova, 1987, 1993a, 1993b, 2002; Kiselyova *et al*., 1998, 2000; Manzanov *et al*., 1987; Pao *et al*., 1999; Savitskii et al., 1979; Villars *et al*., 2001; Yan *et al*., 1994].

It must be pointed out that since there are very many aspects in the domain of the artificial intelligence no criteria for selecting the most suitable algorithm for a particular application were available. After testing many algorithms intended for computer learning applications we formulated the principal criteria of a choice of the programs of computer learning ensuring the most effective decision of chemical tasks:

- possibility of the analysis of the large data volumes;
- possibility of obtaining qualitative classifying regularities at the analysis of the small learning sets;
- automatic exception of properties, which are no important for classification;
- possibility of decision of problems in conditions of weak fulfillment of the principal hypothesis of pattern recognition - hypothesis of compactness;
- fast learning and predicting;
- possibility of analysis of properties with the gaps of some values;
- possibility of analysis of properties having a qualitative nature (e.g., color, type of incomplete electronic shell: s, p, d, or f);
- high accuracy at the decision of chemical tasks;
- convenient interface of the user.

Taking into account these criteria we fixed on the class of algorithms in which all classifying regularities to be found could be presented in the form of a Boolean expression [Gladun, 1995]. This system of concept formation represents information about known chemical systems like - growing pyramidal networks (GPNs). A pyramidal

network is an acyclic oriented graph having no vertices with one entering arc. If the processes of concept formation are determined in the network then the pyramidal network is designated as a growing one [Gladun, 1995]. GPN is built during the process of objects' input. Each object (chemical system) is put in as a set of values of the component properties with an indication of the class to which the system belongs. The nearby values of components' properties are united into one interval using a special program or the experience of a researcher. Concept formation process consists of the analysis of vertices in built network and the choice of those ones that are the most typical for each class [Gladun, 1995]. These vertices became the checking vertices. The resultant concepts (classifying regularities) can be stored in computer memory and printed or read out in the form of a learned GPN or an equivalent Boolean expression, which the values of the component properties make the variables. During the prediction process, the computer receives only the atomic numbers of the elements or designations of simple compounds, while the values of the properties of the appropriate elements or simple compounds are automatically extracted from the DB. They are substituted into the GPN and the researchers can easily obtain the necessary prediction.

Application of Artificial Intelligence to the New Inorganic Materials Computer Design

Using this approach we solved problems of the following types [Kiselyova, 1987, 1993a, 1993b, 2002; Kiselyova *et al*., 1998, 2000; Savitskii *et al*., 1979]:

- prediction of compound formation or non-formation for binary, ternary and more complicated systems;
- prediction of the possibility of forming ternary and more complicated compounds of desired composition;
- prediction of phases with defined crystal structures;
- estimation of phase properties (critical temperature of transition to superconducting state, homogeneity region, etc.).

Shown in Table 1 is a part of the table illustrating predictions of the Heusler phases with composition $ABCu_2$ [Kiselyova, 1987]. These results were obtained in the process of searching for new magnetic materials. All 6 checked predictions agreed with the new experimental data.

Table 1. Part of a table illustrating the prediction of a crystal structure type resembling the Heusler alloys for compounds with the composition $ABCu_2$

A / B	Li	Be	Al	K	Sc	V	Cr	Fe	Co	Ni	Ga	Ge	Y	Nb	Mo	Ru	Rh	Pd
Zn			+									-	-	-	-	-	-	-
Ga	+	+		+	⊕	+	+	+	+	+		O						
In	+	+		+	©	+	+	+	+	+	+	+	©	+	+	+	+	+
Sn	O	-	O	-	-	-		⊕	⊕	⊕	-	-	-	↔	-	-	-	-
Lu	-	-		-	-	-	-	-	-	-		-	-	-	-	-	-	-
Ta		-			-	-	-	-	-	↔		-	-	↔	-	-	-	-
Au	-	-		-	-	-	-	-	-	-		-	-	-	-	-	-	↔
Tl												-						
Pb	-	-	O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Designations:

+ formation of a compound with the composition AB_2Cu and a crystal structure type resembling the Heusler alloys is predicted;
– formation of a compound with a crystal structure type resembling the Heusler alloys is not predicted;
⊕ a compound with a crystal structure type resembling the Heusler alloys was synthesized and appropriate information was used in the computer learning process;
↔ formation of a compound with a crystal structure type resembling the Heusler alloys is known from experiment and appropriate information was used in the computer learning process;
© predicted formation of a compound with a crystal structure type resembling the Heusler alloys which is confirmed by experiment;
O predicted absence of a compound with a crystal structure type resembling the Heusler alloys which is confirmed by experiment; empty square - indeterminate result.

In Table 2 the comparison between the results after predicting the compounds with composition ABX_2 (A and B – various elements; X – S or Se) [Savitskii *et al*., 1979] and the new experimental data. These compounds

were predicted in the process of the search for new semiconductors. Only two predictions were detected to be in error ($CsPrS_2$ and $TlEuSe_2$).

Table 3 shows the predictions of more complicated compounds - new langbeinites with composition $A_2B_2(XO_4)_3$ [Kiselyova *et al*., 2000]. These results are importance for searching for new electro-optical materials. Of 17 checked predictions, 12 agreed with the new experimental data.

Table 2. Part of a table illustrating the prediction of compounds with the composition $A^IB^{III}X_2$

X	S							Se							
A^I B^{III}	Li	Na	K	Cu	Rb	Ag	Cs	Li	Na	K	Cu	Rb	Ag	Cs	Tl
B			ⓒ	ⓒ	ⓒ		ⓒ			+	ⓒ	+	+	+	⊕
Al			ⓒ	⊕	+	⊕	+	ⓒ	⊕	⊕	⊕	+	⊕	+	⊕
Sc	ⓒ	ⓒ	+	⊕	+	+	+	+	+	+	⊕	+	⊕	+	+
Ti	⊕	ⓒ	ⓒ	ⓒ	⊕	+	⊕	ⓒ	ⓒ	+	+	+	+	+	+
V	⊕	ⓒ	+	+	+	+	+	ⓒ	ⓒ	+	+	+	+	+	+
Cr	⊕	⊕	ⓒ	⊕	⊕	⊕	+	+	⊕	+	⊕	⊕	⊕	+	⊕
Mn	+	+	+	+	+	+	+	ⓒ	ⓒ	+	+	ⓒ	+	ⓒ	+
Fe	+	⊕	⊕	⊕	⊕	⊕	⊕	+	+	ⓒ	⊕	ⓒ	⊕	ⓒ	⊕
Co	+	+	+	+	+		+	+	+	+	+	+	+	+	ⓒ
Ni	+	+	+	⊕	+	+	+	+	ⓒ	+	+	+	ⓒ	+	+
Ga	⊕	ⓒ	ⓒ	⊕	ⓒ	⊕	ⓒ	+	+	⊕	⊕	+	⊕	ⓒ	⊕
As	+	⊕	ⓒ	⊕	+	⊕	+	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Y	ⓒ	⊕	ⓒ	⊕	+	ⓒ	+	ⓒ	ⓒ	+	⊕	+	⊕	+	ⓒ
Rh	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
In	ⓒ	⊕	⊕	⊕	⊕	⊕	⊕	ⓒ	ⓒ	⊕	⊕	ⓒ	⊕	+	⊕
Sb	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
La	ⓒ	⊕	⊕	⊕	⊕	↔	⊕		⊕		⊕		↔		↔
Ce	ⓒ	⊕	⊕	⊕	⊕		⊕	ⓒ	⊕	+	⊕		↔	+	+
Pr	⊕	⊕	⊕	⊕	⊕		⊗	+	⊕	+	⊕	ⓒ	↔	+	ⓒ
Nd	⊕	⊕	⊕	⊕	⊕		+	ⓒ	⊕	+	⊕	ⓒ	↔	+	ⓒ
Pm	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+
Sm	⊕	⊕	⊕	⊕	⊕	⊕	+	ⓒ	⊕	+	⊕	ⓒ	↔	+	⊕
Eu	⊕	⊕	⊕	⊕	⊕	⊕	+	+	⊕	+	⊕	+	↔	+	⊗
Gd	⊕	⊕	⊕	⊕	⊕	⊕	+	ⓒ	⊕	+	⊕	ⓒ	ⓒ	+	⊕
Tb	⊕	⊕	⊕	⊕	ⓒ	⊕	+	ⓒ	⊕	+	⊕	ⓒ	⊕	+	ⓒ
Dy	⊕	⊕	⊕	⊕	ⓒ	⊕	+	ⓒ	⊕	+	⊕	+	⊕	+	ⓒ
Ho	⊕	⊕	⊕	⊕	ⓒ	⊕	+	ⓒ	⊕	+	⊕	ⓒ	⊕	+	ⓒ
Er	⊕	⊕	⊕	⊕	ⓒ	⊕	+	ⓒ	⊕	+	⊕	ⓒ	⊕	+	ⓒ
Tm	⊕	⊕	⊕	⊕	ⓒ	⊕	+	+	+	+	⊕	+	⊕	+	ⓒ
Yb	⊕	⊕	⊕	⊕	ⓒ	⊕	+	+	+	+	⊕	+	⊕	+	⊕
Lu	⊕	⊕	⊕	⊕	ⓒ	ⓒ	+	+	+	+	⊕	ⓒ	⊕	+	ⓒ
Tl	+	+	ⓒ	⊕	⊕	ⓒ	⊕	ⓒ		ⓒ	⊕	⊕	⊕	+	

Designations:

\+ predicted formation of a compound with composition ABX_2; - - prediction of no formation of a compound with composition ABX_2;

⊕ compound ABX_2 is known to be formed and this fact is used in the computer learning process;

↔ compound ABX_2 is not known to be formed and this fact is used in the computer learning process;

ⓒ predicted formation of a compound with composition ABX_2 which is confirmed by experiment;

⊗ predicted formation of a compound with composition ABX_2 which is not confirmed by experiment; empty square - indeterminate result.

Predicted compounds were then searched for new magnets, semiconductors, superconductors, electro-optical, acousto-optical, nonlinear optical and other materials required for new technologies. The comparison of these predictions with the experimental data, obtained later, showed that average reliability of predicted compounds exceeds 80%.

Table 3. Part of a table illustrating prediction of a crystal structure type for compounds with the composition $A_2B_2(XO_4)_3$

X	S					Cr					Mo					W				
A / B	Na	K	Rb	Cs	Tl	Na	K	Rb	Cs	Tl	Na	K	Rb	Cs	Tl	Na	K	Rb	Cs	Tl
Mg	L¢	(L)	(L)	(*)	L	L	L©		L©	L©	K¢	(K)	(L)	(L)	(L)		↔	(L)	L©	L
Ca	(*)	(L)	L©	(L)	(*)			L	L	L	(*)	?	?	?	?	(*)	*	?	?	?
Mn	(*)	(L)	(L)	L	(L)	L		(L)	L©	L	K	↔	(L)	(L)	(L)					
Fe	*	L©	L©		(L)	L	K	L	L	L	K	K	?	?	?		K			
Co	(*)	(L)	L©		(L)	L	K	L	L	L		(K)	(L)	(L)	↔		K			
Ni	(*)	(L)	L©	L	L	L		L	L	L	K	(K)	(L)	(L)	(L)					
Cu	(*)		L	*	L	L	K	L	L	L	K	(K)	?	?	?		K			
Zn	*	(L)	L	*	L	L	K	L	L	L	↔	(K)	(K)	-	(K)		K¢			
Sr	(*)	?	?		(*)	*	*	?	?	?	(*)	?	?	?	?	(*)	*	*	*	*
Cd	(*)	(L)	(L)		(L)							K	↔	(L)	K¢	(*)		L	L	L
Ba	(*)		(*)	(*)	(*)	*	*				(*)		*	*	*	*	*			
Pb					*	(*)	*	*©	*	*	(*)	*©	(*)	*¢	*	(*)	*	(*)	(*)	*

Designations:

L formation of a compound with the langbeinite crystal structure type is predicted;

K formation of a compound with the crystal structure type $K_2Zn_2(MoO_4)_3$ is predicted;

– the crystal structure differing from those listed above is predicted;

(L),(K) a compound with corresponding type of crystal structure was synthesized and appropriate information was used in the computer learning process;

↔ a compound with the crystal structure differing from those listed above does not exist at normal conditions and this information was used in the computer learning process;

(*) a compound $A_2B_2(XO_4)_3$ is not formed and this fact was used in the computer learning process;

¢ predicted formation of a compound with this structure type which is not confirmed by experiment;

© predicted formation of a compound with this structure type which is not confirmed by experiment; empty square - indeterminate result.

Information-Analytical System for Materials Computer Design

The promising line of materials computer design is associated with development of information-analytical system [Kiselyova, 1993a, 2002]. This system is intended for data retrieval on known compounds, the prediction of new inorganic compounds, not yet synthesized, and the forecasting of their properties.

This system includes learning and predicting subsystems based on artificial intelligence methods - method of concept formation using growing network CONFOR [Gladun, 1995; Gladun *et al.,* 1995] and other methods of computer learning [http://www.solutions-center.ru]. For increasing reliability of predicting the voting of predictions, which were obtained using various algorithms and component property descriptions will be carried out. The information-analytical system employs also the databases on properties of inorganic compounds, described above, and a DB on chemical elements, a knowledge base, a conversational processor and monitor (Figure 1).

The *knowledge base* of information-analytical system stores the regularities already obtained for various classes of inorganic compounds. They can be use in the prediction of phases and estimation of the phase properties, unless the databases have no such information about the particular chemical systems. Rules in the knowledge base are represented, for example, in the form of growing pyramidal networks, neural networks, logical expressions, etc.

The *conversational processor* manages the conversation of the user with the information-analytical system. It provides an expert in the given application domain a dialog with the information-analytical system also.

In the future, the employment of a linguistic processor in the software or software-hardware support can be expected. It will allow the system to understand the problem-oriented language of the user.

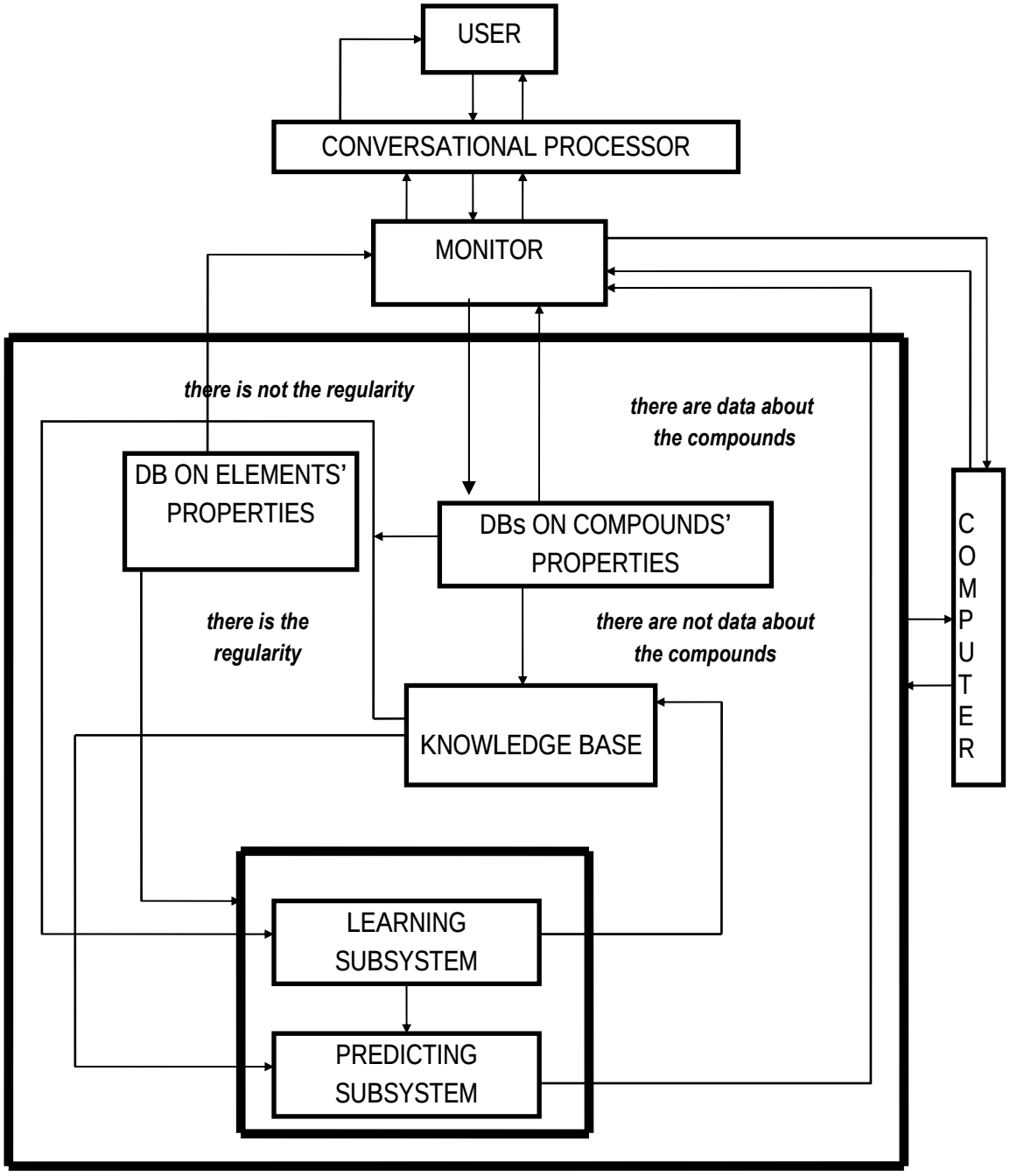


Figure 1. Schematic Diagram of an Information-Analytical System

The *monitor* controls the computation process and provides the interface between the functional subsystems as well as the Internet-access to the system. In addition, the monitor signals whenever new experimental data contradict existing classification regularities. Such contradictions will be eliminated by including the new data in the computer learning and modifying the regularity in the knowledge base.

The information-analytical system operates as follows (Figure 1). The user requests information about a compound of a certain composition. If data about this phase are stored in the databases, they can be extracted for user. If no information about the compound is stored in the databases, or if the information available is incomplete, the computer determines whether the regularity corresponding to the desired property for a compound of a certain type is present in the knowledge base. If the regularity is present, the databases supply the appropriate set of component properties for prediction of the desired characteristic. If the knowledge base does not have the desired regularity, then examples for the computer learning process are extracted from the databases. The correctness of these examples is estimated by the user once more; and, if the samples are found adequate for computer learning, the learning and prediction subsystems process them. The user receives the resultant prediction, while the new classifying regularity is stored in the knowledge base. The above example is the simplest of the problems that can be solved by an information-analytical system. A more complicated problem would be, for example, predicting all possible phases in ternary and multi-component systems, combined with the estimation of their properties. The previous problem can be solved by real-time processing. The latter problem requires much more time.

Conclusion

We have applied artificial intelligence methods for discovering regularities to the design of new inorganic materials. The effectiveness of the proposed approach is illustrated, for example, in Table 1, 2 and 3. The approach discussed in this paper for predicting new materials is based on a process, which is analogous to the search for new materials by an inorganic chemist. It has allowed predicting hundreds of new compounds in ternary, quaternary and more complicated chemical systems, not yet investigated. These compounds are analogs to known substances with important for industry properties. The predictions have been based on the interaction of growing pyramidal networks, computer learning and databases on properties of inorganic compounds. The proposed approach not only simulates the process of design of new inorganic materials but also extends the capabilities of the chemist in discovering new materials with powerful multi-dimensional data analysis tools.

Acknowledgements

Partial financial support from RFBR (Grant N.04-07-90086) is gratefully acknowledged. I should like to thank my colleagues Prof. Victor P.Gladun, Dr.Neonila D.Vashchenko, and Dr.Vitalii Yu.Velichko of the Institute of Cybernetics of the National Academy of Sciences of Ukraine for their help and support.

Bibliography

[Chen et al., 1999] N.Y.Chen, W.C.Lu, R.Chen, P.Qin. Software package “Materials Designer” and its application in materials research. Proc. of Second Int. Conf. Intelligent Processing & Manufacturing of Materials. Honolulu, Hawaii, v.2, July 10-15, 1999.

[Chen et al., 2002] N.Y.Chen, W.C.Lu, C.Ye, G.Li. Application of Support Vector Machine and Kernel Function in Chemometrics. Computers and Applied Chemistry, 2002, v.19:

[Gladun, 1995] V.P.Gladun. Processes of formation of new knowledge. SD "Pedagog 6”, Sofia, 1995 (Russ.).

[Gladun et al., 1995] V.P.Gladun, N.D.Vashchenko. Local Statistic Methods of Knowledge Formation. Cybernetics and Systems Analysis, 1995, v.31.

[Gulyev et al., 1973] B.B.Gulyev, L.F.Pavlenko. Simulation of the search for components of alloy. Avtomatika i Telemekhanika, 1973 v.1 (Russ.).

[http://www.solutions-center.ru] http://www.solutions-center.ru.

[Kiselyova, 1987] N.N.Kiselyova. Prediction of Heusler-phases with composition ABD_2 (D =Co, Ni, Cu, Pd). Izvestiya Akad. Nauk SSSR, Metalli, 1987, v.2 (Russ.).

[Kiselyova, 1993a] N.N.Kiselyova. Information-predicting system for the design of new materials. J.Alloys and Compounds, 1993, v.197.

[Kiselyova, 1993b] N.N.Kiselyova. Prediction of inorganic compounds: experiences and perspectives. MRS Bull., 1993, v.28.

[Kiseleva et al., 1996] N.N.Kiseleva, N.V.Kravchenko, and V.V.Petukhov. Database system on the properties of ternary inorganic compounds (IBM PC version). Inorganic Materials, 1996, v.32.

[Kiselyova et al., 1998] N.N.Kiselyova, V.P.Gladun, N.D.Vashchenko. Computational materials design using artificial intelligence methods. J.Alloys and Compounds, 1998, v.279.

[Kiselyova et al., 2000] N.N.Kiselyova, S.R.LeClair, V.P.Gladun, N.D. Vashchenko. Application of pyramidal networks to the search for new electro-optical inorganic materials. In: IFAC Symposium on Artificial Intelligence in Real Time Control AIRTC-2000. Preprints, Budapest, Hungary, October 2-4, 2000.

[Kiselyova, 2002] N.N.Kiselyova. Computer design of materials with artificial intelligence methods. In: Intermetallic Compounds. Vol.3. Principles and Practice / Ed. J.H.Westbrook & R.L.Fleischer. John Wiley&Sons, Ltd., 2002.

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

[Kutolin et al., 1978] S.A.Kutolin, V.I.Kotyukov. Chemical affinity function and computer prediction of binary compositions and properties of rare earth compounds. Zh.Phys. Chem., 1978, v.52 (Russ.).

[Manzanov et al., 1987] Ye.E.Manzanov, V.I.Lutsyk, M.V.Mokhosoev. Influence of features system selection on predictions of compound formation in systems A_2MoO_4-$B_2(MoO_4)_3$ and A_2MoO_4-$CMoO_4$. Doklady Akad. Nauk SSSR, 1987, v.297 (Russ.).

[Pao et al., 1999] Y.H.Pao, B.F.Duan, Y.L.Zhao, S.R.LeClair. Analysis and visualization of category membership distribution in multivariant data. Proc.Second Int.Conf.Intelligent Processing&Manufacturing of Materials, Honolulu, Hawaii, v.2, July 10-15, 1999.

[Savitskii et al., 1968] E.M.Savitskii, Yu.V.Devingtal, V.B.Gribulya. About recognition of binary phase diagrams of metal systems using computer. Doklady Akad. Nauk SSSR, 1968, v.178 (Russ.).

[Savitskii et al., 1979] E.M.Savitskii, and N.N.Kiselyova. Cybernetic prediction of formation of phases with composition ABX_2. Izvestiya Akad.Nauk SSSR, Neorganicheskie Materialy, 1979, v.15 (Russ.).

[Talanov et al., 1981] V.M.Talanov, L.A.Frolova. Investigation of chalco-spinels formation using method of potential functions. Izvestiya VUZov. Khimiya i Khimicheskaya Technologiya, 1981, v.24 (Russ.).

[Villars et al., 2001] P.Villars, K.Brandenburg, M.Berndt, et al. Binary, ternary and quaternary compound former/nonformer prediction via Mendeleev number. J.Alloys and Compounds. 2001. V.317-318.

[Vozdvizhenskii et al., 1973] V.M.Vozdvizhenskii, V.Ya.Falevich. Application of computer pattern recognition method to identification of phase diagram type of binary metal systems. In: Basic Regularities in Constitution of Binary Metal Systems Phase Diagrams. Nauka, Moscow, 1973 (Russ.).

[Yan, 1994] L.M.Yan, Q.B.Zhan, P.Qin, N.Y.Chen. Study of properties of intermetallic compounds of rare earth metals by artificial neural networks. J.Rare Earths, 1994, v.12.

Author's Information

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

THE DISTRIBUTED SYSTEM OF DATABASES ON PROPERTIES OF INORGANIC SUBSTANCES AND MATERIALS

Nadezhda Kiselyova, Victor Dudarev, Ilya Prokoshev, Valentin Khorbenko, Andrey Stolyarenko, Dmitriy Murat, Victor Zemskov

***Abstract**: The principles of organization of the distributed system of databases on properties of inorganic substances and materials based on the use of a special reference database are considered. The last includes not only information on a site of the data about the certain substance in other databases but also brief information on the most widespread properties of inorganic substances. The proposed principles were successfully realized at the creation of the distributed system of databases on properties of inorganic compounds developed by A.A.Baikov Institute of Metallurgy and Materials Science of the Russian Academy of Sciences.*

***Keywords**: database, distributed information system, inorganic substances and materials, reference database.*

***ACM Classification Keywords**: H.2.4 Distributed databases, H.2.8 Scientific databases.*

Introduction

Now hundreds thousand of inorganic compounds are known. Every year thousands of new substances are added to them. In connection with diversity of applications of inorganic materials, the information on them is scattered over the most various publications. Therefore a search for the information about properties of inorganic compounds, especially if they have been synthesized recently, frequently makes a considerable difficulty and not always it achieves success. A consequence of it is the duplication of investigations on synthesis and research of inorganic substances. In addition, the experts not always can find already synthesized substance that is the most suitable for certain applications. The necessity of acceleration of researches on development and application of new materials were the reasons of creation of numerous databases (DB) on properties of inorganic substances. Thousand of such databases considerably have improved information service for the experts in the field of inorganic chemistry and materials science however there was another problem - problem of a search for DB, in which the needed information on the certain inorganic substances is stored.

Structure of the Distributed System of Databases on Properties of Inorganic Substances and Materials

One of ways of the solution of this problem is the development of some reference database (RDB), which would store the information on where to search for the necessary information on the substance. The distributed system of databases of A.A.Baikov Institute of Metallurgy and Materials Science of the Russian Academy of Sciences (IMET RAS) (fig.1), submitted in the present paper, is a prototype of such information system.

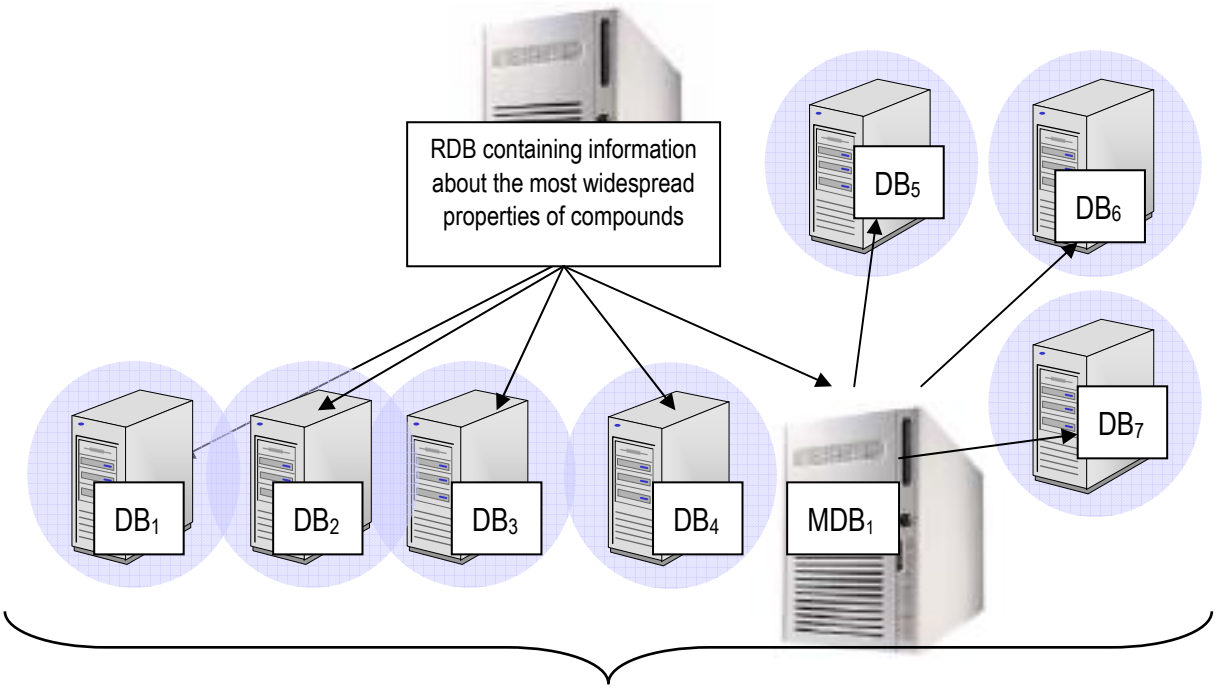


Fig.1. Principles of design of the distributed system of databases on properties of inorganic substances and materials

In this case the DB on properties of inorganic substances «Phases» [Kiseleva et al., 1996], which contains not only data on a site of the various information in other DB but also brief information on the most widespread properties of tens thousand of compounds, for example, melting and boiling points, symmetry of a crystal lattice, etc. (fig.2), carries out the role of a reference database. RDB provides a search for the relevant information

on chemical substances and their properties. The detailed information on substances, which have practical importance, is stored in ordinary DBs, for example, in DBs on properties of materials for electronics [Kiselyova et al., 2004; Khristoforov et al., 2001] developed by us. Thus, the distributed information system, integrated at a level of Web-interfaces, is created.

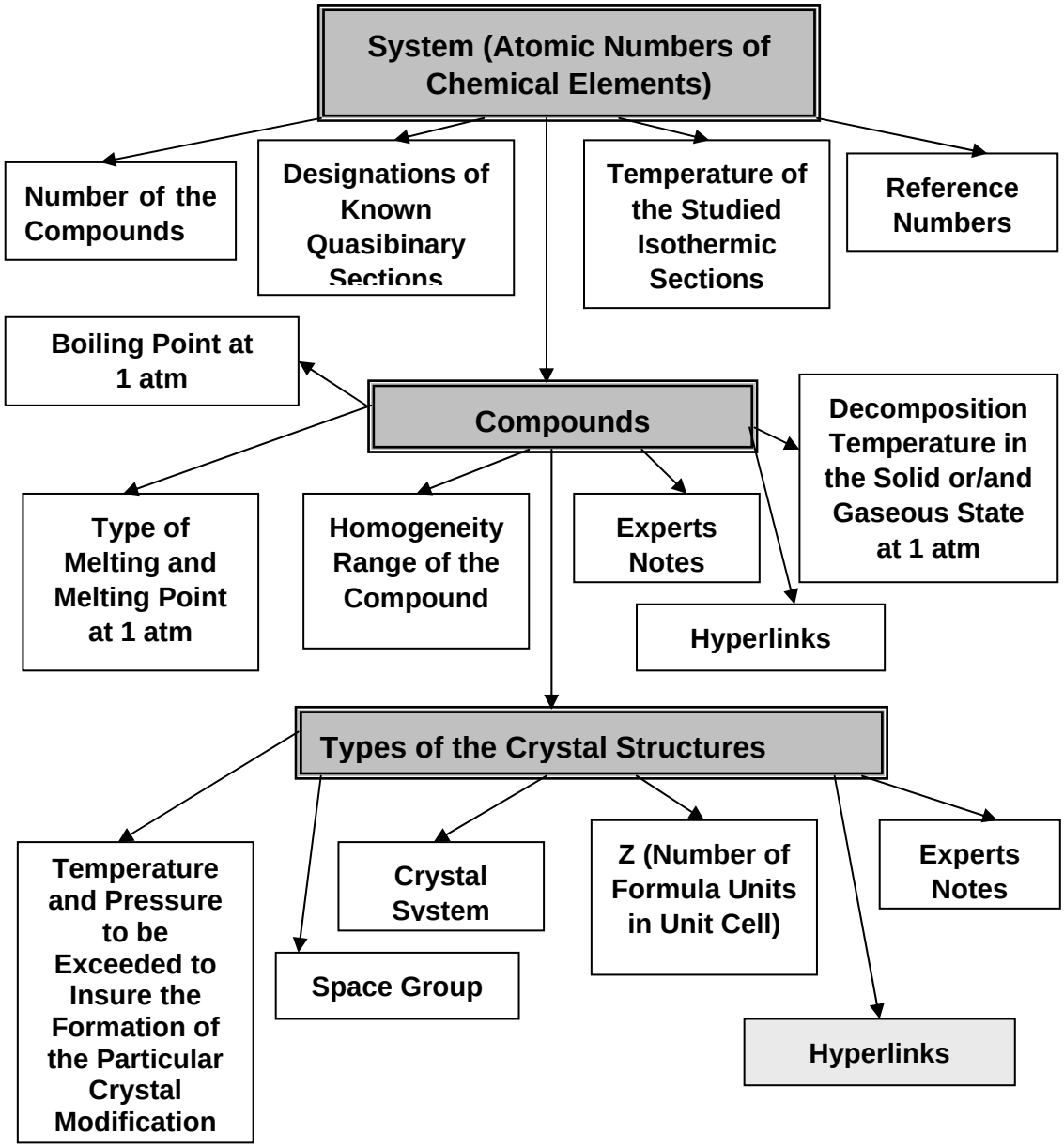


Fig.2. Structure of RDB «Phases»

The concept of design of a special reference database - metabase (MDB) - in the distributed system of the Russian databases on materials for electronics is considered in the terms of the set theory in the paper [Kornyushko et al., 2005]. As shown in this work the search for the relevant information on certain system s can be reduced to definition of the relation R being a subset of Cartesian product, $S \times S$ (in other words, $R \subset S^2$). Here set S is information on substances and systems stored in MDB. The relation R is symmetric at design of the distributed system of DBs on materials for electronics [Kornyushko et al., 2005], since the information of integrated DB mutually supplements each other. Let's note that not always relation R should be strictly symmetric.

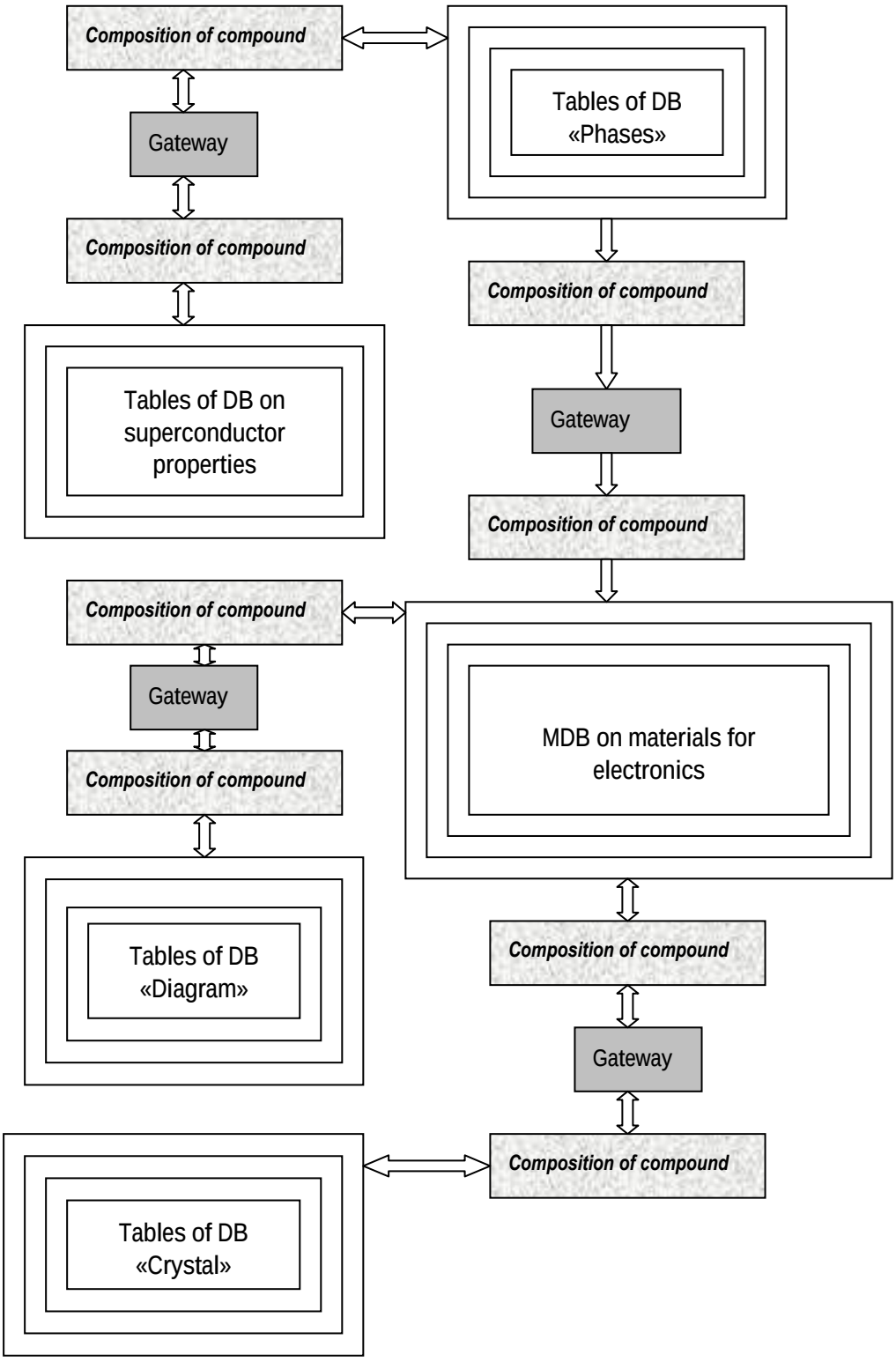


Fig.3. Structure of the distributed system of DBs of IMET RAS

For example, RDB on properties of inorganic substances “Phase” [Kiseleva et al., 1996] contains only brief information on tens thousand of chemical compounds, but specialized DB on properties of acoustic-optical, electro-optical and non-linear optical substances “Crystal” [Kiselyova et al., 2004] or DB on the phase diagrams of semiconductor systems “Diagram” [Khristoforov et al., 2001] contain the detailed information on hundreds substances promising for practical applications. Certainly, the users of the specialized systems own more detailed data in comparison with the information stored in RDB “Phase”. Hence, the users of RDB “Phase”, who search for the relevant information, must have an access to the data in specialized DB, and users, for example, of DB “Diagram” do not have the access to the relevant information on properties of compound from RDB “Phase”.

Hence, in this case relation of relevance, given in [Kornyushko et al., 2005], requires the certain updating. Assume that we have relation N, which describes inadmissible transitions from determined DB into others DB (transitions of a kind $d_1 \to d_2$). Here set D is information on databases. That is if $d_1, d_2 \in D$ and pair $(d_1, d_2) \in N$, the transition from integrated information system d_1 into system d_2 is inadmissible.

For example, if user looks through the information on certain property of compound in one of DB (i.e. actually there is an access to the information determined by a pair (d_1, s_1)), he can have the relevant information on some property of chemical system from another DB, determined by pair (d_2, s_2). As a result, user receives the required new relation of relevance RN as $RN \subset (d_1, s_1) \times (d_2, s_2)$, where $d_1, d_2 \in D; s_1, s_2 \in S$. Thus, the new relation of relevance RN can be constructed on the basis of the old relation R and set N according to the following rule: for any chemical systems $s_1 \in S, s_2 \in S$, if $(s_1, s_2) \in R$ and $d_1, d_2 \notin D$, then $(d_1, s_1), (d_2, s_2) \in R$.

Such decision of a problem of the search for relevant data about properties of substances has many advantages main of which are: simplicity of expansion of the distributed information system, independence on software and hardware platforms, opportunity of actualization and administration of DBs by different organizations which are located in different cities and even in different countries, reduction of traffic, an use of not so powerful, inexpensive servers.

The data on chemical composition (the list of chemical elements and their ratios) are external keys of RDB and various DBs in the distributed system of databases of IMET RAS on properties of inorganic substances and materials (fig.3). It is the most general characteristic of substances, which is inherent in all inorganic objects. Now the distributed system includes besides DB «Phase», in which now the information on properties of ternary compounds is stored, the DB on properties of ternary compounds-superconductors, the DB on properties of acoustic-optical, electro-optical and non-linear optical substances “Crystal” [Kiselyova et al., 2004], DB on phase diagrams of semiconductor systems “Diagram” [Khristoforov et al., 2001] and DB on bandgaps of semiconductors “BandGap”. The latest three DBs, functioning with the use of various software and hardware platforms, were integrated on the basis of the use of special metabase on properties of materials for electronics [Kiselyova et al., 2004; Kornyushko et al., 2005]. Further the distributed system of databases will include other Russian DBs on properties of materials for electronics: DB on intermolecular potentials for components of the CVD processes in microelectronics (Joint Institute of High Temperature of the Russian Academy of Sciences), information system for modeling processes of preparation of epitaxy of hetero-structures of semiconductor materials by the method of liquid epitaxy (M.V.Lomonosov Moscow State Academy of Fine Chemical Technology), etc.

Conclusion

The system of databases of IMET RAS is accessible for the registered users of the Internet: http://www.imet-db.ru.

The work is supported by RFBR, grant №04-07-90086.

Bibliography

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

[Kiseleva et al., 1996] N.N.Kiseleva, N.V.Kravchenko, and V.V.Petukhov. Database system on the properties of ternary inorganic compounds (IBM PC version). Inorganic Materials, 1996, v.32.

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

[Kornyushko et al., 2005] V.Kornyushko, V.Dudarev. Software development for distributed system of Russian databases on electronics materials. Int. J. " Information Theories and Applications", 2006, v.13, n.2, pp.119-124.

Authors' Information

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

Victor A.Dudarev – A.A.Baikov Institute of Metallurgy and Materials Science of Russian Academy of Sciences, programmer, P.O.Box: 119991 GSP-1, 49, Leninskii Prospect, Moscow, Russia, e-mail: vic@osg.ru

Ilya V.Prokoshev – A.A.Baikov Institute of Metallurgy and Materials Science of Russian Academy of Sciences, leading engineer, P.O.Box: 119991 GSP-1, 49, Leninskii Prospect, Moscow, Russia, e-mail: eldream@e-music.ru

Valentin V.Khorbenko – A.A.Baikov Institute of Metallurgy and Materials Science of Russian Academy of Sciences, programmer, P.O.Box: 119991 GSP-1, 49, Leninskii Prospect, Moscow, Russia, e-mail: Khorbenko_v@mail.ru

Andrey V.Stolyarenko – Moscow Institute of Electronics and Mathematics (Technical University), post-graduate student, P.O.Box: 109028, B.Trehsvjatitelsky per. 3/12, Moscow, Russia, e-mail: stol-drew@yandex.ru

Dmitriy P.Murat – Moscow Institute of Electronics and Mathematics (Technical University), post-graduate student, P.O.Box: 109028, B.Trehsvjatitelsky per. 3/12, Moscow, Russia, e-mail: mr_wire@mail.ru

Victor S.Zemskov – A.A.Baikov Institute of Metallurgy and Materials Science of Russian Academy of Sciences, head of Laboratory of Semiconducting Materials, P.O.Box: 119991 GSP-1, 49, Leninskii Prospect, Moscow, Russia, e-mail: zemskov@ultra.imet.ac.ru

TRAINING A LINEAR NEURAL NETWORK WITH A STABLE LSP SOLUTION FOR JAMMING CANCELLATION

Elena Revunova, Dmitri Rachkovskij

***Abstract**: Two jamming cancellation algorithms are developed based on a stable solution of least squares problem (LSP) provided by regularization. They are based on filtered singular value decomposition (SVD) and modifications of the Greville formula. Both algorithms allow an efficient hardware implementation. Testing results on artificial data modeling difficult real-world situations are also provided*

***Keywords**: jamming cancellation, approximation, least squares problem, stable solution, recurrent solution, neural networks, incremental training, filtered SVD, Greville formula*

***ACM Classification Keywords**: I.5.4 Signal processing, G.1.2 Least squares approximation, I.5.1 Neural nets*

Introduction

Jamming cancellation problem appears in many application areas such as radio communication, navigation, radar, etc. [Shirman, 1998], [Ma, 1997]. Though a number of approaches to its solution were proposed [McWhirter, 1989], [Ma et al., 1997], no universal solution exists for all kinds of jamming and types of antenna systems, stimulating further active research to advance existing methods, algorithms and implementations.

Consider an antenna system with a single primary channel and n auxiliary channels. Signal in each channel is, generally, a mixture of three components: a valid signal, jamming, and channel's inherent noise. The problem consists in maximal jamming cancellation at the output while maximally preserving valid signal.

Within the framework of weighting approach [Shirman, 1998], the output is obtained by subtraction of the weighted sum of signals provided by the auxiliary channels from the primary channel signal. The possibility of determining a weight vector $\boldsymbol{w}^*$ that minimizes noise at the output while preserving the valid signal to a maximum degree is, in general case, provided by the following. The same jamming components are present both in primary and auxiliary channels, however, with different mixing factors. Valid signal has small duration and amplitude and is almost absent in auxiliary channels. Characteristics of jamming, channel's inherent noise, and their mixing parameters are stable within the sliding "working window".

These considerations allow us to formulate the problem of obtaining the proper $\boldsymbol{w}^*$ as a linear approximation of a real-valued function $y=f(x)$:

$$F(x) = w_1 h_1(x) + w_2 h_2(x) + \ldots + w_n h_n(x) = \sum_{i=1,\,n} w_i\, h_i\,(x), \qquad (1)$$

where $h_1(x),\ldots,h_n(x)$ is a system of real-valued basis functions; $w_1,\ldots,w_n$ are the real-valued weighting parameters, $F(x)$ is a function approximating $f(x)$.

In our case, $h_1(x),\ldots,h_n(x)$ are signals provided by the n auxiliary channels. Information about $y=f(x)$ at the output of the primary channel is given at discrete set of (time) points $k=1,\ldots,m$ (m is the width of the working window) by the set of pairs $(\boldsymbol{h}^k, y^k)$. It is necessary to find $\boldsymbol{w}^*$ approximating $f(x)$ by $F(x)$ using linear least squares solution:

$$\boldsymbol{w}^* = \mathit{argmin}_{\,w}\, ||\boldsymbol{H}\,\boldsymbol{w} - \boldsymbol{y}||_2, \qquad (2)$$

where $\boldsymbol{H}$ is the so-called $m \times n$ "design matrix" containing the values provided by the n auxiliary channels for all $k=1,\ldots,m$; and $\boldsymbol{y} = [y_1, \ldots, y_m]^T$ is the vector of corresponding y values provided by the primary channel.

After estimating $\boldsymbol{w}^*$, the algorithm's output $\boldsymbol{s}$ is the residual discrepancy:

$$\boldsymbol{s} = \boldsymbol{H}\,\boldsymbol{w}^* - \boldsymbol{y}. \qquad (3)$$

Such a system may be represented as a linear neural network with a single layer of modifiable connections, $n+1$ input and single output linear neurons connected by a weight vector $\boldsymbol{w}$ (Fig. 1). In the case of successful training $\boldsymbol{w}^*$ provides an increased signal-jamming ratio at the output s compared to the input of the primary channel y, at least, for the training set $\boldsymbol{H}$.

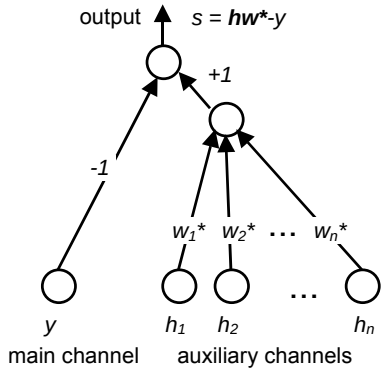


Fig 1. A neural network representation of a jamming canceller

A peculiarity of jamming cancellation problem in such a formulation consists in contamination of both $\boldsymbol{y}$ and $\boldsymbol{H}$ by the inherent noise of channels. Existing algorithms for jamming cancellation in the framework of weighting processing (2)-(3) [Shirman, 1998] do not take into account inherent noise contamination of $\boldsymbol{y}$ and $\boldsymbol{H}$. This results in instability of $\boldsymbol{w}$ estimation, leading, in turn, to a deterioration of cancellation characteristics, and often even to amplification of noise instead of its suppression. Therefore, methods for obtaining $\boldsymbol{w}^*$ should be stable to inherent noise contamination of $\boldsymbol{y}$ and $\boldsymbol{H}$. Other necessary requirements are real-time operation and simplicity of hardware implementation.

Least Squares Solution and Regularization

Generally, the solution of the least squares problem (LSP) (2) is given by

$$\boldsymbol{w}^* = \boldsymbol{H}^+ \boldsymbol{y}; \quad (4)$$

where $\boldsymbol{H}^+$ is pseudo-inverse matrix. If $\boldsymbol{H}$ is non-perturbed (noise is absent), then:

$$for\ m = n,\ rank\ (\boldsymbol{H}) = n = m \Rightarrow det\ \boldsymbol{H} \neq 0.\ \boldsymbol{H}^+ = \boldsymbol{H}^{-1}; \quad (5)$$

$$for\ m > n,\ rank\ (\boldsymbol{H}) = n, \Rightarrow det\ \boldsymbol{H} \neq 0:\ \boldsymbol{H}^+ = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T; \quad (6)$$

$$for\ m = n,\ m > n,\ rank(H) < n \Rightarrow det\ H = 0,\ \boldsymbol{H}^+ = lim_{v \to 0} (\boldsymbol{H}^T\boldsymbol{H} + v^2\boldsymbol{I})^{-1}\boldsymbol{H}^T. \quad (7)$$

$\boldsymbol{H}^+$ for (7) can be obtained numerically using SVD [Demmel, 1997] or the Greville formula [Greville, 1960].

The case when $\boldsymbol{y}$ and elements of matrix $\boldsymbol{H}$ are known precisely is very rare in practice. Let us consider a case that is more typical for jamming cancellation, i.e. when $\boldsymbol{y}$ and $\boldsymbol{H}$ are measured approximately: $\boldsymbol{y} = \boldsymbol{y} + \varsigma$; $\boldsymbol{H} = \boldsymbol{H} + \boldsymbol{\Xi}$; where ς is noise vector, $\boldsymbol{\Xi}$ is noise matrix. In such a case, solutions (5)-(7) may be unstable, i.e. small changes of $\boldsymbol{y}$ and $\boldsymbol{H}$ cause large changes of $\boldsymbol{w}^*$ resulting in instable operation of application systems based on (5)-(7).

To obtain a stable LSP solution, it is fruitful to use approaches for solution of "discrete ill-posed problems" [Hansen, 1998], [Jacobsen et al., 2003], [Reginska, 2002], [Wu, 2003], [Kilmer, 2003]. Such a class of LSPs is characterized by $\boldsymbol{H}$ with singular values gradually decaying to zero and large ratio between the largest and the smallest nonzero singular values. This corresponds to approximately known and near rank-deficient $\boldsymbol{H}$.

Reducing of an ill-posed LSP to a well-posed one by introduction of the appropriate constraints to the LSP formulation is known as regularization [Hansen, 1998]. Let us consider a problem known as standard form of the Tikhonov regularization [Tikhonov, 1977]:

$$argmin_{\ w} \{||\boldsymbol{y} - \boldsymbol{H}\,\boldsymbol{w}||_2 + \lambda\, ||\boldsymbol{w}||_2\}. \quad (8)$$

Its solution $\boldsymbol{w}_\lambda$ may be obtained in terms of SVD of $\boldsymbol{H}$ [Hansen, 1998]:

$$\boldsymbol{w}_\lambda = \Sigma_{i=1,\,n}\ f_i\ \boldsymbol{u}_i^T\boldsymbol{y}\ /\sigma_i\ \boldsymbol{v}_i; \quad (9)$$

$$f_i = \sigma^2_i/(\sigma^2_i + \lambda^2), \quad (10)$$

where σ_i are singular values, $\boldsymbol{u}_1 \ldots \boldsymbol{u}_n$, $\boldsymbol{v}_1 \ldots \boldsymbol{v}_n$ are left and right singular vectors of $\boldsymbol{H}$, f_i are filter factors.

Note that solution of (8) using truncated SVD method [Demmel, 1997] is a special case of (9), (10) with $f_i \in \{0,1\}$.

Algorithmic Implementation of Solutions of Discrete ill-posed LSPs

Requirements of an efficient hardware implementation of jamming cancellation pose severe restrictions on the allowable spectrum of methods and algorithms. In particular, methods of $\boldsymbol{w}^*$ estimation are required to allow for parallelization or recursion. Taking this into account, let us consider some algorithmic implementations of (8).

SVD-based solution. The implementation of SVD as a systolic architecture with paralleled calculations is considered in [Brent, 1985]. We have developed a systolic architecture that uses effective calculation of $\boldsymbol{u}_i$, σ_i, and $\boldsymbol{v}_i$ for obtaining the regularized solution $\boldsymbol{w}_\lambda^*$ (9)-(10). The architecture implements two regularization techniques: truncated and filtered SVD [Hansen, 1998].

Advantages of SVD-based solution are accuracy and parallelism. Drawbacks are connected with the hardware expenses for calculation of trigonometric functions for diagonalization of sub-matrices and implementation of systolic architecture itself.

Solution based on the Greville formula and its modifications. Let us consider another stable method for $\boldsymbol{w}^*$ estimation based on the Greville formula, which can be readily implemented in hardware because of its recursive character. A recursive procedure for the LSP solution [Plackett, 1950] for a full-rank $\boldsymbol{H}$ is as follows:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \boldsymbol{b}_{k+1}(y_{k+1} - \boldsymbol{h}^T_{k+1}\boldsymbol{w}_k); \quad k = 0, 1, \ldots, \tag{11}$$

$$\boldsymbol{b}_{k+1} = \boldsymbol{P}_k \boldsymbol{h}_{k+1}/(1 + \boldsymbol{h}^T_{k+1}\boldsymbol{P}_k \boldsymbol{h}_{k+1}); \tag{12}$$

$$\boldsymbol{P}_{k+1} = (\boldsymbol{H}_{k+1}{}^T \boldsymbol{H}_{k+1})^{-1} = (\boldsymbol{I} - \boldsymbol{b}_{k+1}\boldsymbol{h}^T_{k+1})\boldsymbol{P}_k; \tag{13}$$

where $\boldsymbol{h}_k$ is the kth row (sample) of $\boldsymbol{H}$; $\boldsymbol{P}_0 = 0$; $\boldsymbol{w}_0 = 0$. Note that this provides an iterative version of training algorithm for a neural network interpretation of Fig. 1.

The Greville formula [Greville, 1960] allows $\boldsymbol{b}_{k+1}$ calculation for (11) without $(\boldsymbol{H}_{k+1}{}^T \boldsymbol{H}_{k+1})^{-1}$ calculation, thus overcoming the problem of rank-deficiency of $\boldsymbol{H}$:

$$\boldsymbol{b}_{k+1} = (\boldsymbol{I} - \boldsymbol{H}^+_k \boldsymbol{H}_k)\boldsymbol{h}_{k+1} / \boldsymbol{h}^T_{k+1}(\boldsymbol{I} - \boldsymbol{H}^+_k \boldsymbol{H}_k)\boldsymbol{h}_{k+1}; \text{ if } \boldsymbol{h}^T_{k+1}(\boldsymbol{I} - \boldsymbol{H}^+_k \boldsymbol{H}_k)\boldsymbol{h}_{k+1} \neq 0; \tag{14}$$

$$\boldsymbol{b}_{k+1} = \boldsymbol{H}^+_k (\boldsymbol{H}^+_k)^T \boldsymbol{h}_{k+1} / (1 + \boldsymbol{h}^T_{k+1}\boldsymbol{H}^+_k (\boldsymbol{H}^+_k)^T \boldsymbol{h}_{k+1}); \text{ if } \boldsymbol{h}^T_{k+1}(\boldsymbol{I} - \boldsymbol{H}^+_k \boldsymbol{H}_k)\boldsymbol{h}_{k+1} = 0; \tag{15}$$

$$\boldsymbol{H}^+_{k+1} = (\boldsymbol{H}^+_k - (\boldsymbol{b}_{k+1}\boldsymbol{h}^T_{k+1}\boldsymbol{H}^+_k \mid \boldsymbol{b}_{k+1})). \tag{16}$$

$\boldsymbol{w}^*$ obtained by (11)-(13) using (14)-(16) is equivalent to $\boldsymbol{w}^*$ obtained by (7) for precisely specified $\boldsymbol{H}$. Presence of $\boldsymbol{H}^+_k$ and $\boldsymbol{H}_k$ in (14)-(16) makes recursion more resource- and computation-expensive than (12)-(13). As a new sample arrives, it is necessary to calculate $\boldsymbol{H}^+_k\boldsymbol{H}_k$ or $\boldsymbol{H}^+_k(\boldsymbol{H}^+_k)^T$ that requires calculation of $\boldsymbol{H}^+_k$ and storage of $\boldsymbol{H}^+_k$ and $\boldsymbol{H}_k$. These drawbacks are overcome by an improvement of the Greville formula proposed recently in [Zhou, 2002].

For $\boldsymbol{h}^T_{k+1}\boldsymbol{Q}_k = 0$ calculations of $\boldsymbol{b}_{k+1}$ and $\boldsymbol{P}_{k+1}$ are made by (12)-(13). If $\boldsymbol{h}^T_{k+1}\boldsymbol{Q}_k \neq 0$

$$\boldsymbol{b}_{k+1} = \boldsymbol{Q}_k\boldsymbol{h}_{k+1}/(\boldsymbol{h}^T_{k+1}\boldsymbol{Q}_k\boldsymbol{h}_{k+1}); \tag{17}$$

$$\boldsymbol{P}_{k+1} = (\boldsymbol{I} - \boldsymbol{b}_{k+1}\boldsymbol{h}^T_{k+1})\boldsymbol{P}_k(\boldsymbol{I} - \boldsymbol{b}_{k+1}\boldsymbol{h}^T_{k+1})^T + \boldsymbol{b}_{k+1}\boldsymbol{b}^T_{k+1}; \tag{18}$$

$$\boldsymbol{Q}_{k+1} = (\boldsymbol{I} - \boldsymbol{b}_{k+1}\boldsymbol{h}^T_{k+1})\boldsymbol{Q}_k. \tag{19}$$

Here $\boldsymbol{P}_k = \boldsymbol{H}^+_k(\boldsymbol{H}^+_k)^T$ is Hermitian $n \times n$ matrix; $\boldsymbol{P}_0 = 0$, $\boldsymbol{Q}_k = \boldsymbol{I} - \boldsymbol{H}^+_k \boldsymbol{H}_k$; $\boldsymbol{Q}_0 = \boldsymbol{I}$.

We further modified the Greville formula so that $\boldsymbol{w}_{k+1}$ is equivalent to the regularized solution $\boldsymbol{w}_\lambda{}^*$ (9). This is achieved by comparison of vector norm $\boldsymbol{h}^T_{k+1}\boldsymbol{Q}_k$ not with 0, but with some threshold value 0_{eff} calculated from noise matrix $\boldsymbol{\Xi}$. We name such an algorithm "pseudo-regularized modification of the Greville formula" (PRMGF).

The algorithm (11)-(13), (17)-(19) calculates $\boldsymbol{w}^*$ using all previous samples. However, for a non-stationary case it is necessary to process only a part of the incoming samples inside a sliding working window. Full recalculation of $\boldsymbol{H}^+_{k+1}$ for estimation of $\boldsymbol{w}_{k+1}$ as each new sample arrives can be avoided by using inverse recurrent representation of [Kirichenko, 1997]. For the purpose of removing the row $\boldsymbol{h}^T_1$ from $\boldsymbol{H}_{k+1}$, $\boldsymbol{H}^+_k$ is represented through $\boldsymbol{H}^+_{k+1} = (\boldsymbol{b}_1 | \boldsymbol{B}_{k+1})$ as follows.

For a linear independent row:

$$(\boldsymbol{H} + \boldsymbol{h}\boldsymbol{e}^T)^+ = \boldsymbol{H}^+ - \boldsymbol{H}^+\boldsymbol{h}\boldsymbol{h}^T\boldsymbol{Q}/(\boldsymbol{h}^T\boldsymbol{Q}\boldsymbol{h}) - \boldsymbol{Q}\boldsymbol{e}\boldsymbol{e}^T\boldsymbol{H}^+/\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} + \boldsymbol{Q}\,\boldsymbol{e}\boldsymbol{h}^T\,\boldsymbol{Q}(\boldsymbol{H}^T)\,(1 + \boldsymbol{e}^T\boldsymbol{H}^+\boldsymbol{h})/\,\boldsymbol{h}^T\,\boldsymbol{Q}(\boldsymbol{H}^T)\boldsymbol{h}\,\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e}; \tag{20}$$

and for a linear dependent row:

$$(\boldsymbol{H} + \boldsymbol{h}\boldsymbol{e}^T)^+ = (\boldsymbol{I} - \boldsymbol{z}\boldsymbol{z}^T/\,||\boldsymbol{z}\,||^2)\,\boldsymbol{H}^+; \; \boldsymbol{z} = \boldsymbol{H}^+\boldsymbol{h} - \boldsymbol{e}/||\boldsymbol{e}||^2. \tag{21}$$

Thus, we propose to use PRMGF with a sliding window for the case, when it is required to obtain $\boldsymbol{w}^*$ not for the whole training set, but for its subset of a fixed size. For initial $k < m$ samples $\boldsymbol{h}_k$, $\boldsymbol{w}^*_k$ is recursively calculated by PRMGF. For $k > m$, (20)-(21) are used for updating $\boldsymbol{H}^+$ by removing the sample that has left a working window, and the incoming sample s is taken into account using PRMGF as earlier.

Advantages of PRMGF with a sliding window include:

- natural embedding into recursive algorithm for $\boldsymbol{w}^*$;

- increase of calculation speed due to using $\boldsymbol{h}^T_{k+1}$ instead of $\boldsymbol{H}_k$, also resulting in reduction of required memory;
- additional memory reduction since $\boldsymbol{P}_k$, $\boldsymbol{K}_k$ and $\boldsymbol{Q}_k$ have fixed $n \times n$ dimension for any k;
- further increase of calculation speed when sliding window is used due to the Greville formula inversion;
- considerably smaller hardware expenses in comparison with SVD;
- $\boldsymbol{w}^*$ close to the Tikhonov regularized solution for noisy, near rank-deficient $\boldsymbol{H}$ (at least, for small matrices);
- natural interpretation as an incrementally trained neural network.

Example of Modeling a Jamming Cancellation System

Let's compare the following jamming cancellation algorithms: ordinary LS-based (6); non-truncated SVD-based [Demmel, 1997]; truncated SVD-based (9) with $f_i = \{0,1\}$; PRMGF-based (section 3.2). We use near rank-deficient $\boldsymbol{H}$, which is critical for traditional jamming cancellation algorithms – e.g., for ordinary LS-based ones.

Testing scheme and cancellation quality characteristics. In a real situation, all antenna channels receive jamming signals weighted by the gain factor that is determined by the antenna directivity diagram in the direction of particular jamming. We simulated signals in antenna channels as follows:

$$\boldsymbol{X} = \boldsymbol{S}\,\boldsymbol{M} + \boldsymbol{\Xi}; \tag{22}$$

where $\boldsymbol{X}$ is $L \times (n+1)$ matrix of signals in antenna channels ($\boldsymbol{H}$ is sub-matrix of $\boldsymbol{X}$); L is the number of samples; n is the number of auxiliary channels; $\boldsymbol{S}$ is jamming signals' matrix; $\boldsymbol{\Xi}$ is channel inherent noise matrix; $\boldsymbol{M}$ is mixing matrix.

Jamming signals and channels' inherent noise are modeled by normalized centered random variables with normal and uniform distribution correspondingly. $\boldsymbol{M}$ is constructed manually, values of its elements are about units, rank deficiency was achieved by entering identical or linearly dependent rows. For ideal channels without inherent noise, rank deficiency of $\boldsymbol{M}$ gives rise to strict rank deficiency of $\boldsymbol{H}$. Inherent noise results in near rank-deficient $\boldsymbol{H}$. Tests were carried out for n=8 auxiliary channels.

The main characteristics of jamming cancellers are: jamming cancellation ratio (K^c) and jamming cancellation ratio vs inherent noise level in auxiliary channels K^{naux}: $K^c = f(K^{naux})$ [Bondarenko, 1987] at fixed inherent noise at the primary channel K^{n0}.

$$K^c = P^{in}/P^{out}, \tag{23}$$

where P^{in} and P^{out} is power of jamming in the primary channel and in the output of jamming canceller, respectively. In all tests, the valid signal with amplitude not exceeding amplitude of primary channel jamming was present at the input for 5 nearby samples. L=1000; m=16; $K^{n0} = \{0.1, 0.2, 0.3\}$, $K^{n0} >> K^{naux}$ to complicate the algorithm's operation.

Testing results. A family of jamming cancellation characteristics $K^c = f(K^{naux})$ for rank-deficient $\boldsymbol{M}$ and near rank-deficient $\boldsymbol{H}$ is shown in Fig.2. K^{naux} varied from $1.6\ 10^{-9}$ up to $6.4\ 10^{-6}$. K^c for the ordinary LS did not exceed 1 at $K^{naux} < 2.5\ 10^{-8}$. For truncated SVD and PRMGF $K^c \approx 10$ ($K^{n0} = 0.1$) are nearly constant over the whole range of K^{naux} and close to each other. Note that for a full-rank matrix $\boldsymbol{H}$, K^n for all algorithms was approximately the same and large in the considered range of K^{naux}.

It may seem from the analysis of the shown results that one may use the ordinary LS algorithm at increased level of K^{naux}. However, roll-of of cancellation characteristic is also observed when jamming intensity in auxiliary channels is much more than the inherent noise level. To show that, let us consider $K^c(P^{in})$ for near rank-deficient $\boldsymbol{H}$ (Fig.3). Jamming power P^{in} changed from $9 \cdot 10^3$ to $1.2 \cdot 10^7$ by step $2 \cdot 10^2$, $K^{naux} = 0.1$. For $P^{in} > 10^4$, K^c for ordinary LS and non-truncated SVD decreases. For truncated SVD and PRMGF, $K^c \approx 9$ ($K^{n0} = 0.1$) are constant and close to each other. In this case, we cannot artificially increase inherent noise level because it will completely mask the valid signal.

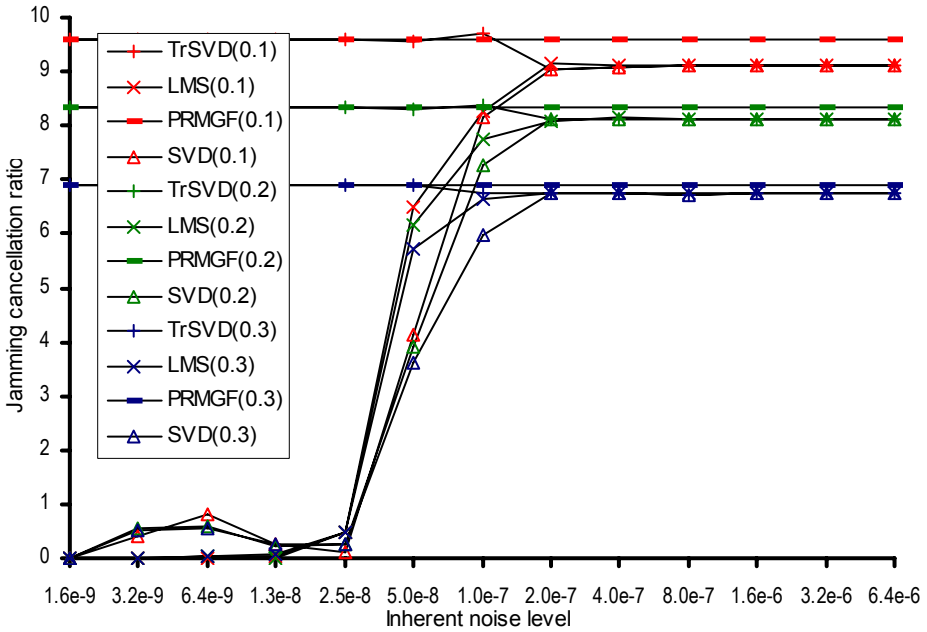


Fig. 2. Kc= f(Knaux) for near rank-deficient H

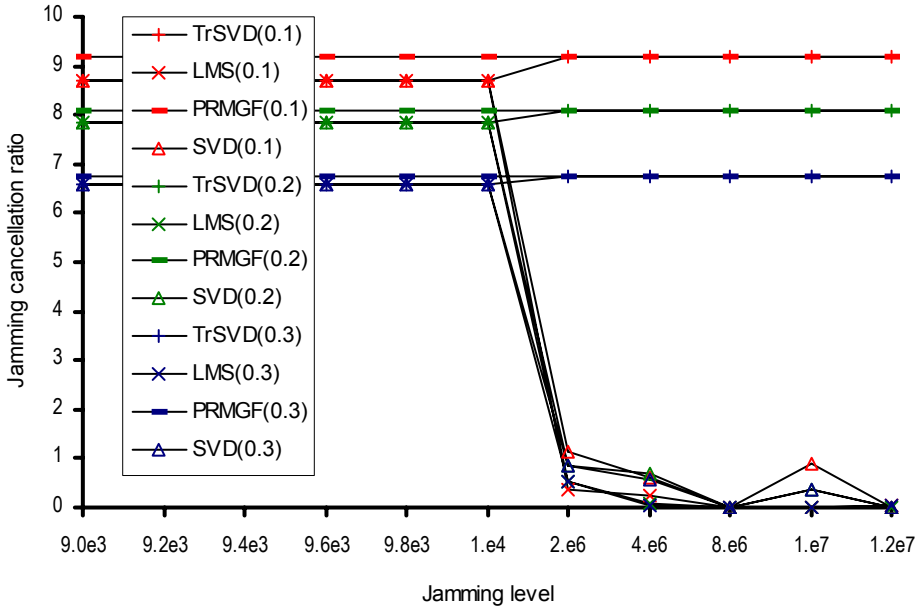


Fig. 3. Kc(Pin) for near rank-deficient H

Conclusions

In the framework of this work, two new jamming cancellation algorithms have been developed based on the so-called weighting approach. Special requirements to the problem have resulted in its classification as a discrete ill-posed problem. That has allowed us to apply an arsenal of the regularization-based methods for its stable solution - estimation of weight vector $\boldsymbol{w}^*$.

The standard form of Tikhonov regularization based on SVD has been transformed to efficient hardware systolic architecture. Besides, pseudo-regularized modification of the Greville formula allowed us to get weight vector estimations very close to estimations for a truncated SVD based regularization - at least for $\boldsymbol{H}$ of about tens of columns. Testing on near rank-deficient $\boldsymbol{H}$ has shown that distinctions in $\boldsymbol{w}^*$ obtained by both algorithms are

of the order 10^{-5}. A combined processing technique based on a regularized modification of the Greville formula and inverse recurrent representation of Kirichenko permits a more efficient processing of data for a sliding working window.

Testing on artificial data that model real-world jamming cancellation problem has shown an efficient cancellation for near rank-deficient $\boldsymbol{H}$. For the developed PRMGF-based algorithm the jamming cancellation ratio is near constant and considerably higher than 1 in the whole range of variation of auxiliary channels' inherent noise and jamming amplitude. On the contrary, for the non-regularized LS method the ratio roll-offs to less than 1, meaning jamming amplification.

A straightforward neural network interpretation of such a system is provided. The developed algorithms and computer architectures for their implementation can be applied to solution of other discrete ill-posed LS problems and systems of linear algebraic equations.

Bibliography

[Bondarenko, 1987] B.F. Bondarenko, Bases of radar systems construction, Kiev, 1987 (in Russian).

[Brent, 1983] R.P. Brent, H. T. Kung and F. T. Luk, Some linear-time algorithms for systolic arrays, in Information Processing 83, 865–876, North-Holland, Amsterdam, 1983.

[Brent, 1985] R.P. Brent and F. T. Luk, The solution of singular-value and symmetric eigenvalue problems on multiprocessor arrays, SIAM J. Scientific and Statistical Computing 6, 69–84, 1985.

[Demmel, 1997] J.W. Demmel, Applied Numerical Linear Algebra. SIAM, Philadelphia, 1997.

[Greville, 1960] T. N. E. Greville, Some applications of the pseudoinverse of a matrix, SIAM Rev. 2, pp. 15-22, 1960.

[Hansen, 1998] P.C. Hansen, Rank-deficient and discrete ill-posed problems. Numerical Aspects of Linear Inversion, SIAM, Philadelphia, 1998.

[Jacobsen et al., 2003] M. Jacobsen, P.C. Hansen and M.A. Saunders, Subspace preconditioned LSQR for discrete ill-posed problems, BIT Numerical Mathematics 43: 975–989, 2003.

[Kilmer, 2003] M. Kilmer, Numerical methods for ill-posed problems, Tufts University, Medford, MAPIMS Workshop, 2003.

[Kirichenko, 1997] N.F. Kirichenko, Analytical representation of pseudo-inverse matrix perturbation, Cybernetics and system analysis, 2, pp. 98-107, 1997 (in Russian).

[Ma et al., 1997] J. Ma, E.F. Deprettere and K.K. Parhi, Pipelined cordic based QRD-RLS adaptive filtering using matrix look ahead. In Proc. IEEE Workshop on Signal Processing Systems, pp.131-140 Lecester, UK, November, 1997.

[Ma, 1997] J. Ma, Pipelined generalized sidelobe canceller, Technique Report, Circuits and Systems, Delft University of Technology, 1997.

[McWhirter, 1989] J.G. McWhirter and T.J. Shepherd, Systolic array processor for MVDR beamforming, In IEEE Proceedings vol.136, pp. 75-80, April 1989.

[Plackett, 1950] R. L. Plackett, Some theorems in least squares, Biometrika, 37, pp. 149-157, 1950.

[Reginska, 2002] T. Reginska, Regularization of discrete ill-posed problems, IM PAN Preprints, 2002, http://www.impan.gov.pl/Preprints.

[Shirman, 1998] J.D. Shirman, Radio-electronic systems: bases of construction and the theory, Moscow, 1998 (in Russian).

[Tikhonov, 1977] A.N. Tikhonov, V.Y. Arsenin, Solution of ill-posed problems. V.H. Winston, Washington, DC, 1977.

[Wu, 2003] L. Wu, A parameter choice method for Tikhonov regularization, Electronic Transactions on Numerical Analysis, Volume 16, pp. 107-128, 2003.

[Zhou, 2002] J. Zhou, Y. Zhu, X. R. Li, Z. You, Variants of the Greville formula with applications to exact recursive least squares. SIAM J. Matrix Anal. Appl. Vol.24, No.1, pp.150-164, 2002.

Authors' Information

Elena G. Revunova, Dmitri A. Rachkovskij - International Research and Training Center of Information Technologies and Systems, Pr. Acad. Glushkova 40, Kiev 03680, Ukraine. email: helab@i.com.ua, dar@infrm.kiev.ua

APPLIED PROBLEMS OF FUNCTIONAL HOMONYMY RESOLUTION FOR RUSSIAN LANGUAGE

Olga Nevzorova, Julia Zin'kina, Nicolaj Pjatkin

***Abstract**: Applied problems of functional homonymy resolution for Russian language are investigated in the work. The results obtained while using the method of functional homonymy resolution based on contextual rules are presented. Structural characteristics of minimal contextual rules for different types of functional homonymy are researched. Particular attention is paid to studying the control structure of the rules, which allows for the homonymy resolution accuracy not less than 95%. The contextual rules constructed have been realized in the system of technical text analysis.*

***Keyword**: natural language processing, functional homonymy, resolution of homonymy*

***ACM Classification Keywords**: H.3.1.Information storage and retrieval: linguistic processing*

Introduction

There is no common opinion on the phenomenon of homonymy in linguistic literature. The content of the phenomenon, principles of classification, classificatory schemes are being discussed. The most common classification divides homonyms into lexical ones, i.e. referring to the same part of speech, and grammatical ones, i.e. referring to different parts of speech. A term "functional homonyms" is also spread in linguistic literature. This term, offered by O.S. Ahmanova, has been accepted in the works by V.V. Babajceva [Babajceva, 1967] and her followers. V.V. Babajceva defines functional homonyms as "words with homophony, historically coming from one word family, referring to different parts of speech" and spreads the occurrence of functional homonymy not only on autosemantic parts of speech, but also on synsemantic ones. In the actual work methods of automatical resolution of functional homonymy of different types are researched.

Success of applied research in computer linguistics depends remarkably on the availability of appropriate linguistic resources, lexicographical ones being the most important. In the recent years dictionaries of homonyms of Russian language by different authors have been published. In these dictionaries, the phenomenon of homonymy has been represented with various degree of fullness. Thus, for example, in N.P. Kolesnikov's dictionary [Kolesnikov, 1978] the phenomenon of homonymy is understood in an extended sense, homoforms, homophones and homographs being included into the circle of the occurrences studied besides lexical homonyms. Attempts on describing functional homonyms have been made in separate homonym dictionaries by O.S. Ahmanova [Ahmanova, 1984], O.M. Kim [Kim et al., 2004]. The appearance of an Internet-resource by N.G. Anoshkina [Anoshkina, 2001], attempting to gather all grammatical homonyms, stimulated further development of theoretical and applied research, including that on the classification of functional homonymy types. The grand problem is mismatch of grammatical descriptions of homonyms in these dictionaries. For example, the comparison of the grammatical descriptions of 560 homonyms terminating on letter 'o' in [1-4] have shown that only three homonyms have been described with the same grammatical features.

In the work [Kobzareva et al., 2002] a classification of 58 homonymy types is given. This classification served as a base for working out rules of contextual resolution of functional homonymy. The rules are offered in the article. In the course of research some changes have been made to the basic classification of functional homonymy types. The changes were connected with the appearance of new subtypes, addition and exclusion of some types. It is obvious that it needs to develop new dictionary of functional homonyms on the basis of representative corpus of Russian texts for applied research.

Method of Contextual Resolution of Functional Homonymy (Problems of Building up Contextual Rules)

Theoretical research on the problem of functional homonymy resolution in texts has a long history. At the end of the 50-s in works by K.E. Harper [Harper, 1956], A. Caplan [Caplan, 1955], studying and describing contextual conditions in which some or another meaning of the word would be realized was accepted as the main way of homonymy resolution. Either the surroundings of the word in the text or the words with which the word given was used would be implied under "context". The question of minimal resolving context was also actual for the researches. Noteworthy are the results obtained by A. Caplan [Caplan, 1955] in his research on minimal resolving context. About 140 frequently used polysemantic English words (lexical homonyms mainly) in different contextual surroundings have been analyzed in the work.

The following types of contexts have been selected:

1). Combination with the preceding word – P1.

2). Combination with the following word – F1.

3). Combination with both preceding word and following word – B1.

4). Combination with two preceding words – P2.

5). Combination with two following words – F2.

6). Combination with two preceding words and two following words – B2.

7). The whole sentence – S.

The main conclusion implied that B1 chain was more productive as regards the effect of reducing polysemy (the proportion of the quantity of word meanings in concrete context to their number in zero context) than P2 and F2 contexts and would be almost equal to the effect given by S. Another conclusion emphasized the importance of material context type. That is, whether autosemantic parts of speech or so-called "particles" (including prepositions, conjunctions, auxiliary verbs, articles, pronouns, adverbs like *there* etc) exist in the direct surroundings of the word. The context with autosemantic parts of speech gives much better results than that with "particles". General conclusions by A. Caplan imply that the most useful context is the one consisting of one word to the left and one word to the right from the word given. If one word of the context is a particle, the context should be extended to two words on both sides.

Despite numerous references to the results quoted above in Western literature of the 60-s, their practical usage for Russian language in real contexts is hardly possible. The real situation with the resolution of functional and lexical homonymy in Russian language is far more complicated and cannot be resolved on the basis of simplified rules. In the work [Kobzareva et al., 2002], a special dictionary of diagnostic situations (DDS) has been developed. These situations assign linear structure description of the minimal context necessary for the identification of the homonym's part of speech.

DDS situations consist of two parts:

- component chain of the sentence (may be discontinuous) marks the situation;

- conditions put on the markers and their surroundings, defining the meaning of the homonym.

While analyzing the examples given in the article [Kobzareva et al., 2002], it is remarkable that the borders of the minimal resolving context become moveable. Symbols belonging to a certain multitude (varying for different homonymy types) act as borders. Besides, the authors allow for discontinuous contexts. This complicates the situation even further. Besides, yet another thing should be pointed out. With a set of rules used for the resolution of some homonymy type one of the main problems is building up a control structure for the order of the usage.

Saying it another way, the method of contextual resolution of functional homonymy offered by the authors of the present article includes:

- Establishing a full classification of functional homonymy types.

- Selecting the minimal set of resolving contexts for each type.

The minimality of the set means that for every functional homonymy type it should be estimated, how difficult it would be to recognize each part of speech belonging to the type. Then the set of resolving contexts (SRC) with minimal difficulty of resolution should be built up. The algorithmic form of this demand will look as follows:

If a rule from SRC has been applied to a functional homonym of T1 or T2 type, then the type of the homonym is defined by the rule applied, else the alternative type is given to X.

- Building up a control structure for the SRC, allowing for the maximum resolution accuracy.

Classification of the Functional Homonymy Types

The ground for basic classification of functional homonymy types was the classification offered in the article [Kobzareva et al., 2002]. It had been built up on the basis of N.G. Anoshkina's dictionary of grammatical homonyms [Anoshkina, 2001]. The classification proposes 58 types of homonyms, the first ten types being the most numerous (the whole number of homonyms is 2965). Less than 5 homonyms are included into each of 26 other types. In spite of the undoubted importance of the classification, it requires further development. The latter concerns the addition of new types as well as additional subdivision of types and creation of subtypes. For example, a new type "short form of adjective # category of state" like *gotovo (*in Russian*)/ ready*. Second kind of expansions is connected, for instance, with three subtypes in Vf/N* type, where Vf stands for a verb form and N* = {N - a noun, Npr - pronominal noun}.

The example in Russian is:

1) bereg N - *a brink* *2) bereg (inf. berech')* V, Past Tense - *saved*

The subtypes have been selected on the basis of a syntactic criterium, i.e. the recognition of their representatives requires working out remarkably different rules and control structures (the examples are in Russian):

<Vf/N*>$_1$: *pojmu* = 1) N - *bottom-land*; 2) V, Future Tense - *I will understand*;
<Vf/N*>$_2$: *I'nu* = 1) N – *flax*; 2) V, Present Tense - *I am clinging to sth*;
<Vf/N*>$_{3.1}$: *stali* = 1) V, Past Tense - *we became*; 2) N - *steel*;
<Vf/N*>$_{3.2}$: *zarosli* = 1) V, Past Tense - *became overgrown*; 2) N – *bush.*

Rules of Resolving Functional Homonymy of Some Types (Realization)

In this part, an example of resolving functional homonymy of type N*/Comp, here N* ={N =a noun, Npr=pronominal noun}, Comp – comparative. Let the following system of denotations be introduced:

X – functional homonym; P– preposition. The term $X_{\bigcap_{pgn} N^*}$ means that X is complied with N* by the grammatical

characteristics given (p – case, g – gender, n – singular/plural).

A Z may be present in the rule. It means that an inserted construction of some type may be present here. In Russian language, the discontinuance of a linear sequence because of such constructions is possible almost everywhere in the sentence. We have used a typology of such constructions allowing to identify them. For example, constructions expressing feelings and emotions – '*k schast'ju' (in Russian)/ fortunately* –, one expressing the degree of trustworthiness – '*nesomnenno' (in Russian)/ assuredly* – etc. The recognition of them is a major point, as punctuation marks belong to the multitude of context restrainers. Consequently, recognizing the function of the punctuation mark is important. That is, we should know the one separating an inserted construction from all the others, as it is not considered a context constraint.

While working out the rules of resolution of functional homonymy of different types we have offered a method of multiplex resolution of functional homonymy. It allows to resolve groups of homogeneous homonyms. These rules are particularly typical for some certain types of homonymy, like N*/A* type (A*={A=full adjective form, Av=participle, Apr=pronominal adjective}), D/Abr type, D/Abr/Vsp type (D – adverb, Abr - the short form of the adjective, Vsp - predicate noun) and others.

Let us exemplify the group of contextual rules for N*/Comp type. The minimal set of resolving contexts was selected for resolving an homonym as Comp. The control structure of the general rule defines the order of usage and the results (in rectangles on the scheme).

$$1)\quad if\quad [\frac{P\cap_X (Z)\overline{N}\cap_p}{\le 3}X]\quad then\quad X=N*$$

$$2)\quad if\quad [\frac{A*\cap_{pgn} x(Z)\overline{N}\cap_{A*}}{\le 3}X]\quad then\quad X=N*$$

$$3)\quad if\quad [\frac{X(Z)N*_{p2}}{\le 5}]\ /\ [\frac{X(Z)\overline{,}A*_{p2}}{\le 4}]\quad then\quad X=Comp$$

$$4)\quad if\quad [\frac{X\ ,than}{\le 4}]\ /\ [\frac{X\ ,that}{\le 4}]\ /\ [\frac{that\ X}{\le 3}]\quad then\quad X=Comp$$

$$5)\quad if\quad [\frac{V_{SV}(Z)X}{\le 2/\otimes}]\quad then\quad X=Comp$$

$$6)\quad if\quad [\frac{all\ X}{\le 2/\otimes}]\quad then\quad X=Comp$$

$$7)\quad if\quad [Comp\ \exists < S_{\&} > X]\ /[X\ \exists < S_{\&} > Comp]\quad then\quad X=Comp$$

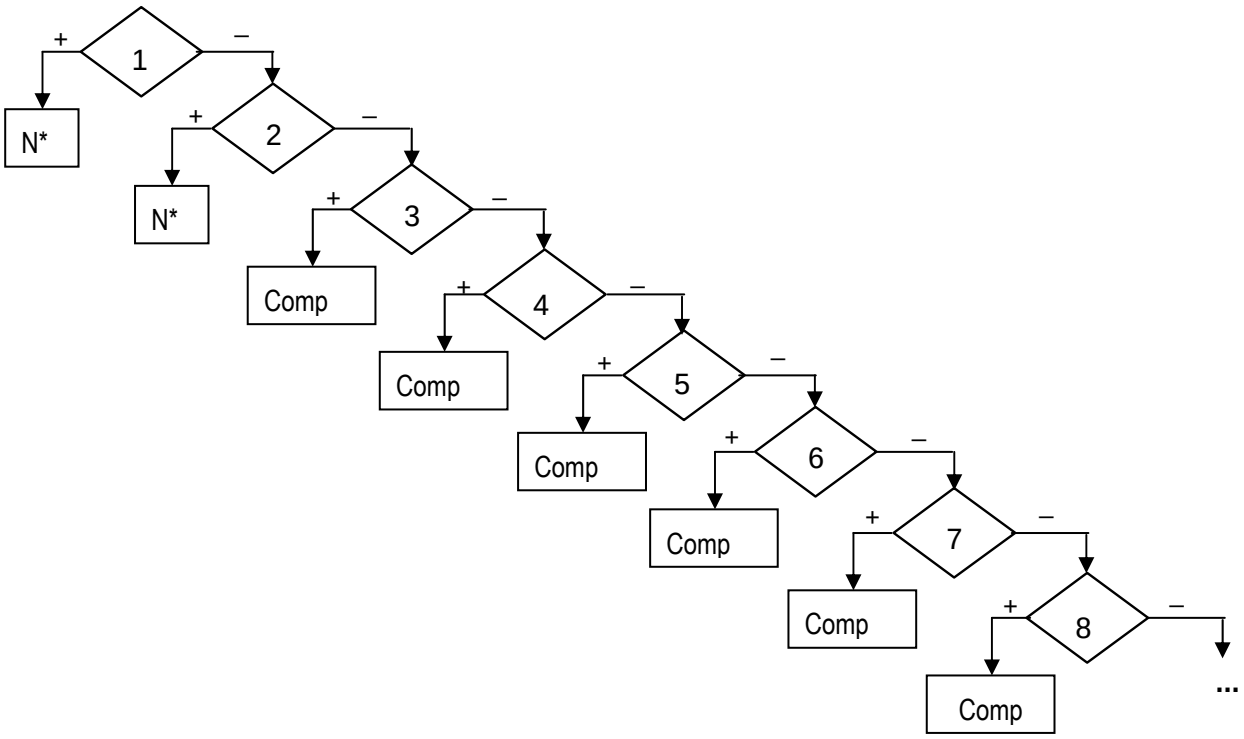
$$8)\quad if\quad [\frac{D_{md}(Z)X}{\le 2/\otimes}]\quad then\quad X=Comp$$

$$9)\quad if\quad length(\otimes\frac{X}{\le 4}\otimes)\quad then\quad X=Comp$$

$$10)\quad if\quad [\frac{V_f(Z)X}{\le 2/\otimes}]\ /\ [\frac{X(Z)V_f}{\le 2/\otimes}]\quad then\quad X=Comp$$

$$else\quad X=N*$$

Picture 1. Rules of resolution of functional homonymy of N*/Comptype



Picture 2. The control structure of the general N*/Comp rule

Conclusion

Program realization of syntactical processing module of technical texts analysis system is currently being completed. The module is to comprise the process of resolution of functional homonymy. Real technical texts, however, contain not all of the types. There are no types with interjections, for example, or with stylistically substandard words. Other limitations are connected with the vocabulary of technical texts itself (there are very few imperative verb forms, for instance). Nevertheless, contextual rules allow for such verbs to be recognized. Module setting operations allow for the amount of homonymy types liable to resolution to be changed.

Testing of the program module for the resolution of functional homonymy has given good results on the types realized. For some types, the accuracy of resolution is 100%, in the worst cases it is not less than 95%. The reasons for erroneous situations' appearance are accidental concord in the context analyzed, context insufficiency or resource insufficiency (absence of a case frame dictionary for different parts of speech). Some mistakes in the resolution can be sorted out in the course of further analysis.

Acknowledgements

The work has been completed with partial support of Russian Foundation Basic Research (grant № 05-07-90257).

Bibliography

[Ahmanova, 1984] A.S. Ahmanova. Slovar omonimov russkogo jazyka. M., 1984. (In Russian).

[Kolesnikov, 1978] N.P. Kolesnikov. Slovar omonimov russkogo jazyka. Tbilisi, 1978. (In Russian)

[Anoshkina, 2001] J.G. Anoshkina. Slovar omonimichnyh slovoform russkogo jazyka. M: Mashinnyj fond russkogo jazyka Instituta russkogo jazyka RAN, 2001. (In Russian) (http://irlras-cfrl.rema.ru:8100/homoforms/index.htm).

[Caplan, 1955] A. Caplan. An Experimental Study of Ambiguity and Context // Mech. Translation, vol. 2, No 2, Nov. 1955.

[Harper, 1956] K.E. Harper. Contextual Analysis // Mech. Translation, vol. 4, No 3, Dec. 1956.

[Kobzareva et al., 2002] T.U. Kobzareva, R.N. Afanasiev. Universalnyj modul predsintaksicheskogo analiza omonimii chastej rechi b RJa na osnove slovarja diagnosticheskih situacij // Trudy mezhdunar. konferencii Dialog'2002. M., 2002. S. 258-268. (In Russian).

[Kobzareva et al., 2002] O.M. Kim, I.E. Ostrovkina. Slovar graamaticheskih omonimov russkogo jazyka. M., 2004. (In Russian).

[Babajceva, 1967] V.V. Babajceva. Perehodnye konstrukcii v sintaksise. Voronezh, 1967. (In Russian).

Authors' Information

Olga Nevzorova - Research Institute of Mathematics and Mechanics, Kazan State Pedagogical University, Kazan, Russia; e-mail: olga.Nevzorova@ksu.ru

Julia Zin'kina - Kazan State University

Nicolaj Pjatkin - Research Institute of Mathematics and Mechanics, Kazan, Russia; e-mail: nikolaip@mail.ru

AN APPROACH TO COLLABORATIVE FILTERING BY ARTMAP NEURAL NETWORKS

Anatoli Nachev

***Abstract**: Recommender systems are now widely used in e-commerce applications to assist customers to find relevant products from the many that are frequently available. Collaborative filtering (CF) is a key component of many of these systems, in which recommendations are made to users based on the opinions of similar users in a system. This paper presents a model-based approach to CF by using supervised ARTMAP neural networks (NN). This approach deploys formation of reference vectors, which makes a CF recommendation system able to classify user profile patterns into classes of similar profiles. Empirical results reported show that the proposed approach performs better than similar CF systems based on unsupervised ART2 NN or neighbourhood-based algorithm.*

***Keywords**: neural networks, ARTMAP, collaborative filtering*

***ACM Classification Keywords**: I.5.1 Neural Nets*

Introduction

The World Wide Web has been established as a major platform for information and application delivery. The amount of content and functionality available often exceeds the cognitive capacity of users. This problem has also been characterized as information overload [13]. Since the World Wide Web has become widespread, more and more applications exist that are suitable for the application of social information filtering techniques. Recommender systems are now widely used in e-commerce applications to assist customers to find relevant products from the many that are frequently available. Collaborative filtering is a key component of many of these systems, in which recommendations are made to users based on the opinions of similar users in a system.

In collaborative filtering preferences of a user are estimated through mining data available about the whole user population, implicitly exploiting analogies between users that show similar characteristics.

A variety of CF filters or recommender systems have been designed, most of which can be grouped into two major classes: memory-based and model-based [10].

Memory-based algorithms maintain a database of all users' known preferences for all items, and for each prediction, perform some computation across the entire database. This approach is simpler, seem to work reasonably well in practice, and new data can be added easily and incrementally, however, it can become computationally expensive in terms of both time and space complexity, as the size of the database grows.

On the other hand, model-based CF algorithms use the users' preferences to learn a model, which is then used for predictions. They are small, fast, and essentially as accurate as memory based methods. Memory requirements for the model are generally less than for storing the full database and predictions can be calculated quickly once the model is generated.

This paper presents a model based-approach to collaborative filtering by using supervised ARTMAP neural network. Proposed algorithm is based on formation of reference vectors that make a CF system able to classify user profile patterns into classes of similar profiles, which forms the basis of a recommendation system.

Related Work

A variety of collaborative filters or recommender systems have been designed and deployed. The Tapestry system relied on each user to identify like-minded users manually [5]. GroupLens [6] and Ringo [7], developed independently, were the first CF algorithms to automate prediction. Both are examples of the more general class of memory-based approaches, where for each prediction, some measure is calculated over the entire database of users' ratings. Typically, a similarity score between the active user and every other user is calculated. Predictions

are generated by weighting each user's ratings proportionally to his or her similarity to the active user. A variety of similarity metrics is possible. Resnick et al. [6] employ the Pearson correlation coefficient. Shardanand and Maes [7] test a few metrics, including correlation and mean squared difference. Breese et al. [8] propose the use of vector similarity, based on the vector cosine measure often employed in information retrieval. All of the memory-based algorithms cited predict the active user's rating as a similarity-weighted sum of the others users' ratings, though other combination methods, such as a weighted product, are equally plausible. Basu et al. [9] explore the use of additional sources of information (for example, the age or sex of users, or the genre of movies) to aid prediction. Breese et al. [8] identify a second general class of model-based algorithms. In this approach, an underlying model of user preferences is first constructed, from which predictions are inferred. The authors describe and evaluate two probabilistic models, which they term the Bayesian clustering and Bayesian network models.

Adaptive Resonance Theory

Adaptive Resonance Theory (ART) [1] [2] is family of neural networks for fast learning, pattern recognition, and prediction, including both unsupervised: ART1, ART2, ART2-A, ART3, Fuzzy ART, Distributed ART; and supervised: ARTMAP, Fuzzy ARTMAP, ART-EMAP, ARTMAP-IC, ARTMAP-FTR, Distributed ARTMAP, and Default ARTMAP systems.

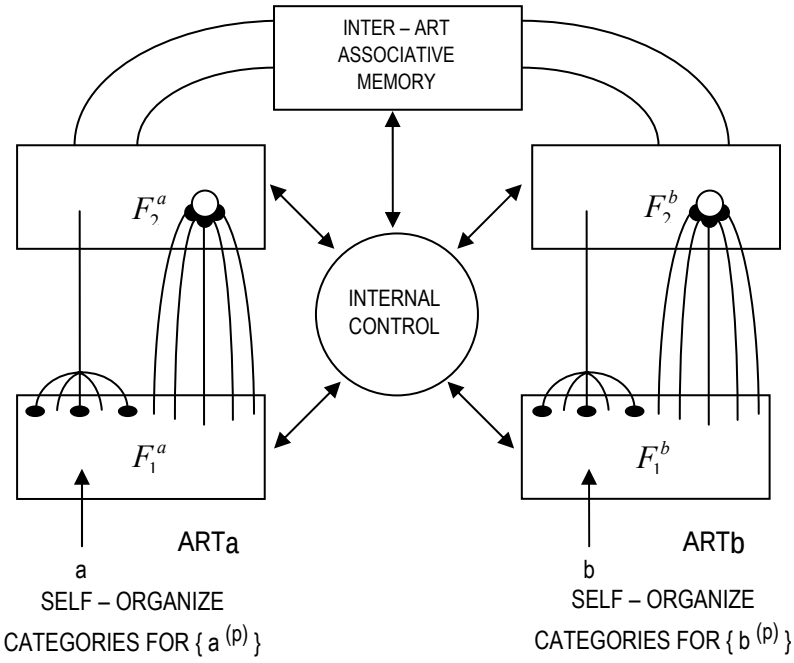


Figure 1. Components of an ARTMAP system.

These ART models have been used for a wide range of applications, such as remote sensing, medical diagnosis, automatic target recognition, mobile robots, and database management. ART1 self-organizes recognition codes for binary input patterns; ART2 does the same for analogue input patterns. ART3 is the same as ART2 but includes a model of the chemical synapse that solves the memory-search problem of ART systems.

Any ART module consists of two fields, F_1 and F_2, connected by two sets of adaptive connections: bottom-up connections, $F_1 \to F_2$; and top-down connections $F_2 \to F_1$. In an ART module, the input pattern is presented to the F_1 field which normalizes and contrast-enhances features of the pattern. F_2 activation is then calculated by multiplying the F_1 pattern with the bottom-up weights. Lateral inhibition in the F_2 field then finds a winning F_2 node. The degree of match between the top-down expectation pattern of the winning F_2 node and the F_1

pattern is then evaluated in a vigilance test to determine whether it is sufficient. If it is, then learning occurs in both the top-down and bottom-up connections of the winning F_2 node, otherwise the winning F_2 node is reset and the search continues. ARTMAP is a supervised neural network which consists of two unsupervised ART modules, ART_a and ART_b and an inter-ART associative memory, called a map-field (see Figure 1).

ARTMAP Network

ARTMAP architectures are neural networks that develop stable recognition codes in real time in response to arbitrary sequences of input patterns. They were designed to solve the stability-plasticity dilemma that every intelligent machine learning system has to face: how to keep learning from new events without forgetting previously learned information. ARTMAP networks were designed to accept binary input patterns [3].

An ART module has three layers: the input layer (F_0), the comparison layer (F_1), and the recognition layer (F_2) with m , m and n neurons, respectively (see module ART_a or ART_b in Figure 2). The neurons, or nodes, in the F_2 layer represent input categories. The F_1 and F_2 layers interact with each other through weighted bottom-up and top-down connections, which are modified when the network learns. There are additional gain control signals in the network that regulate its operation.

At each presentation of a non-zero binary input pattern x $(x \in \{0,1\}, i = 1,2,\ldots,m)$, the network attempts to classify it into one of its existing categories based on its similarity to the stored prototype of each category node. More precisely, for each node j in the F_2 layer, the bottom-up activation

$$T_j = \sum_{i=1}^{m} x_i Z_{ij}$$

is calculated, where Z_{ij} is the strength of the bottom-up connection between F_1 node i and F_2 node j. Since both the input and the bottom-up weight vectors are binary with Z_{ij} being the normalized version of z_{ij} , T_j , can also be expressed as

$$T_j = \left|x \cap Z_j\right| = \frac{\left|x \cap z_j\right|}{\beta + \left|z_j\right|} \tag{1}$$

where $|.|$ is the norm operator ($|x| \equiv \sum_{i=1}^{m} x_i$), z_j is the binary top-down template (or prototype) of category j , and $\beta > 0$ is the choice parameter. Then the F_2 node J that has the highest bottom-up activation is selected, i.e. $T_j = max\{T_j \mid j = 1,2,\ldots,n\}$. The prototype vector of the winning node $J(z_J; z_{Ji} \in \{0,1\}, i = 1,2,\ldots,m)$ is then sent down to the F_1 layer through the top-down connections, where it is compared to the current input pattern: the strength of the match is given by

$$\frac{\left|x \cap z_J\right|}{|x|},$$

which is compared with a system parameter ρ called vigilance $(0 < \rho \le 1)$. If the input matches sufficiently, i.e., the match strength $\ge \rho$, then it is assigned to F_2 node J and both the bottom-up and top-down connections are adjusted for this node. If the stored prototype z_J does not match the input sufficiently (match strength $< \rho$), the winning F_2 node J is reset for the period of presentation of the current input. Then another F_2 node (or category) will be selected, whose prototype will be matched against the input. This "hypothesis-testing" cycle is repeated until the network either finds a stored category whose prototype matches the input closely enough, or allocates a new F_2 node. Then learning takes place as described above. After an initial period of self-stabilization, the network will directly (i.e., without search) access the prototype of one of the categories it has

found in a given training set. The higher the vigilance level, the larger number of smaller, or more specific, categories will be created. If $\rho = 1$, the network will learn every unique input perfectly with a different category.

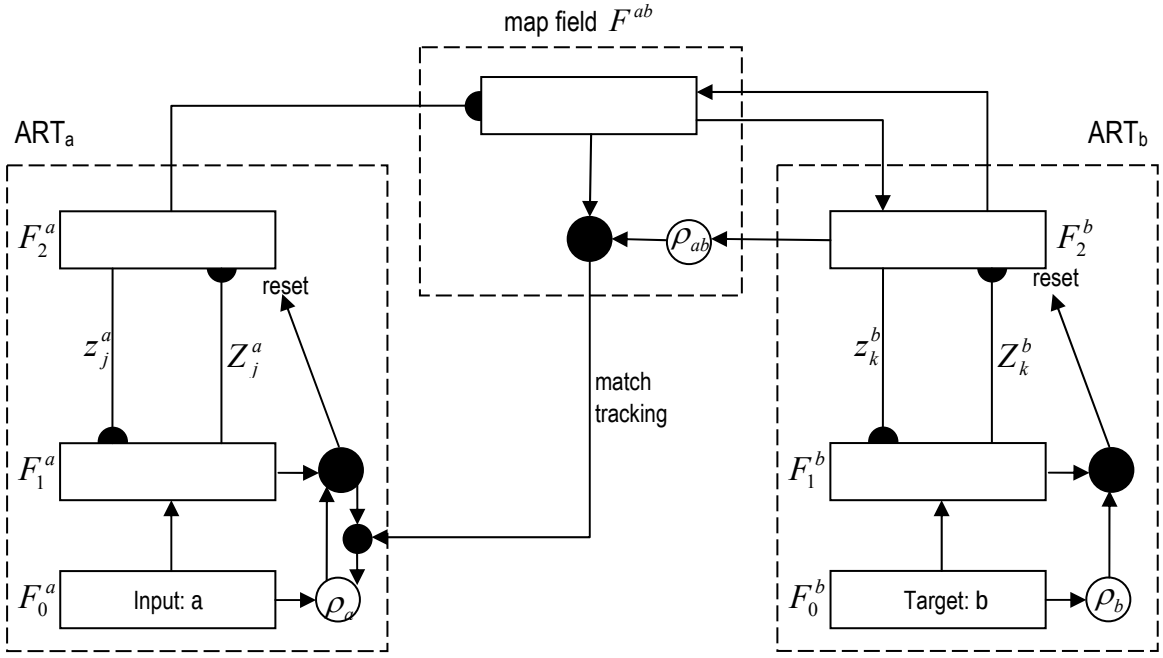


Figure 2. Architecture of ARTMAP network.

The architecture of the ARTMAP network can be seen in Figure 2. It consists of two ART modules that are linked together through an inter-ART associative memory, called map field F^{ab}. Module ART$_a$ (with a baseline vigilance $\overline{\rho}_a$ learns to categories input patterns presented at layer F_0^a, while module ART$_b$ with vigilance ρ_b develops categories of target patterns presented at layer F_0^b. Modules F_2^a and F^{ab} are fully connected via associative links whose strengths are adjusted through learning. There are one-to-one, two-way, and non-modifiable connections between nodes in the F^{ab} and F_2^b layers, i.e., each F_2^b node is connected to its corresponding F^{ab} node, and vice versa. A new association between an ART$_a$ category J and an ART$_b$ category K is learned by setting the corresponding $F_2^a \to F^{ab}$ link to one and all other links from the same ART$_a$ node to zero. When an input pattern is presented to the network, the F^{ab} layer will receive inputs from both the ART$_a$ module through the previously learned $J \to K$ associative link and the ART$_b$ module from the active F_2^b category node. If the two F^{ab} inputs match, i.e., the network's prediction is confirmed by the selected target category, the network will learn by modifying the prototypes of the chosen ART$_a$ and ART$_b$ categories according to the ART learning equations shown above. If there is a mismatch at the F^{ab} layer, a map field reset signal will be generated, and a process called match tracking will start, whereby the baseline vigilance level of the ART$_a$ module will be raised by the minimal amount needed to cause mismatch with the current ART$_a$ input at the F_1^a layer. This will subsequently trigger a search for another ART$_a$ category, whose prediction will be matched against the current ART$_b$ category at the F^{ab} layer again. This process continues until the network either finds

an ART_a category that predicts the category of the current target correctly, or creates a new F_2^a node and a corresponding link in the map field, which will learn the current input/target pair correctly. The ART_a vigilance is then allowed to return to its resting level $\bar{\rho}_a$.

After a few presentations of the entire training set, the network will self-stabilize, and will read out the expected output for each input without search.

ARTMAP Learning

All ART1 learning is gated by F_2 activity - that is - the adaptive weights z_{Ji} and Z_{iJ} can change only when the J^{-th} F_2 node is active. Then both $F_2 \to F_1$ and $F_1 \to F_2$ weights are functions of the F_1 vector x, as follows:

Top-down learning

Stated as a differential equation, this learning rule is [3]

$$\frac{d}{dt} z_{ji} = y_j (x_i - z_{ji}) \tag{2}$$

In equation (2), learning by z_{ji} is gated by y_j. When the y_j gate opens - that is when $y_j > 0$ - then learning begins and z_{ji} is attracted to x_i. In vector terms, if $y_j > 0$, then approaches x. Initially all z_{ji} are maximal: $z_{ji}(0) = 1$. Thus with fast learning, the top-down weight vector z_J is a binary vector at the start and end of each input presentation. F_1 activity vector can be described as

$$x = \begin{cases} I & if\ F_2\ is\ inactive \\ I \cap z_J & if\ the\ J^{th}\ F_2\ node\ is\ inactive \end{cases} \tag{3}$$

When node J is active, learning causes

$$z_J(new) = I \cap z_J(old) \tag{4}$$

where $z_J(old)$ denotes z_J at the start of the input presentation.

Bottom-up learning

In simulations it is convenient to assign initial values to the bottom-up $F_1 \to F_2$ adaptive weights Z_{ij} in such a way that F_2 nodes first become active in the order $j = 1, 2, \ldots$. This can be accomplished by letting $Z_{ij}(0) = \alpha_j$, where $\alpha_1 > \alpha_2 > \ldots > \alpha_N$. Like the top-down weight vector z_J, the bottom-up $F_1 \to F_2$ weight vector $Z_J \equiv (Z_{1J}, Z_{2J}, \ldots, Z_{iJ}, \ldots, Z_{MJ})$ also becomes proportional to the F_1 output vector x when the F_2 node J is active. In addition, however, the bottom-up weights are scaled inversely to $|x|$, so that

$$Z_{iJ} \to \frac{x_i}{\beta + |x|} \quad \text{where } \beta > 0 .$$

This $F_1 \to F_2$ learning realizes a type of competition among the weights z_J adjacent to a given F_2 node J. This competitive computation could alternatively be transferred to the F_1 field, as it is in ART2 [2]. During learning

$$Z_J(new) = \frac{I \cap z_J(old)}{\beta + |I \cap z_J(old)|} \tag{5}$$

The Z_{ij} initial values are required to be small enough so that

$$0 < \alpha_j = Z_{ij}(0) < \frac{1}{\beta + |I|} \qquad \text{for all } F_0 \to F_1 \text{ inputs } I .$$

Experiments

A series of experiments were conducted to estimate ARTMAP architecture as a model-based approach to CF. For experiments a CF component, based on ARTMAP neural network was used. It was designed with 60 F_2^a neurons, 40 F_2^b neurons, and 40 F^{ab} map-field neurons. Two other CF components were also used – one based on ART2 network with 60 F_2 neurons and one memory-based CF component that incorporates the popular neighbourhood-based algorithm, as described in [4].

Most of the results presented here were obtained by using the publicly available EachMovie dataset [12]. It contains 2,811,983 ratings on a scale from 1 to 5 for 1,628 movies by 72,916 users. On average, each user rated about 46.3 movies. As in [4], analysis was restricted to the users who have minimum the average for the database rating activity (45 entries) in their profile, and extracted 196817 vote records of the first 2000 of those users from the database. Restricted number of user reveals the performance of the model-based CF approach under conditions where the ratio of users to items is low. This is condition that every CF service has to go through in its first phase.

The resulting dataset of users and their votes was divided into two data sets - a training set that contains randomly selected 60 rated items, and a test set with randomly selected 40 rated items. To simulate a growing database, three experiments were conducted using 30%, 60% and 100% of available profile entries, with 40 control set entries in each case that we used to evaluate the computed recommendations. The three different subsets have been used as training sets for the neural networks and as input for the memory based method. Afterwards 1, 5, 15, and 30 recommendations were computed and compared to the control set of 40 profile entries.

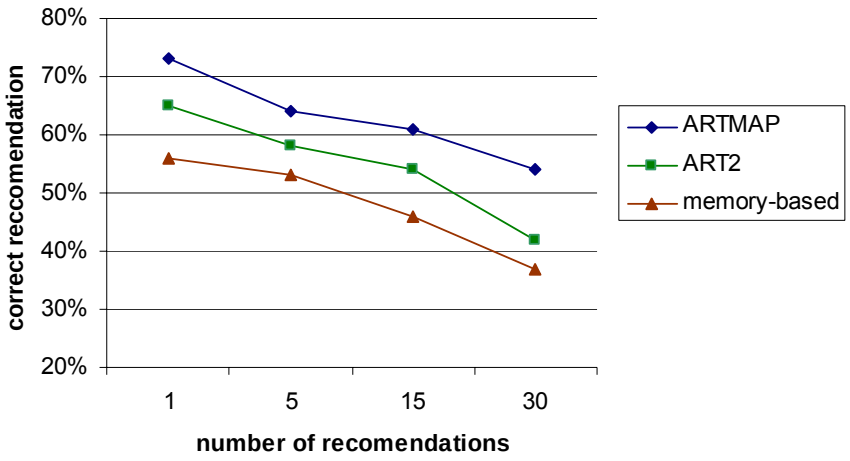


Figure 3. Correct recommendations with growing dataset.

Figure 3 summarizes the results of those experiments. It can be seen that in terms of correct recommendations in conditions of growing dataset the ARTMAP network performed better than both ART2 network and memory based method.

Second group of experiments aimed to compare response time of both the ART2 NN and memory-based neighborhood algorithm. Five series of experiments were conducted with growing number of users. The four test sets contain profile entries for 500, 1000, 1500 and 2000 of the user data set. Each time recommendations were computed, the response time has been measured. Results summarized in Figure 4 show that the proposed ARTMAP CF component performs better that both ART2 and model-based components in terms of response time when the number of users increases. As expected and shown in Figure 4, the number of users has a much less significant influence to the performance of the neural network based methods than the memory-based one.

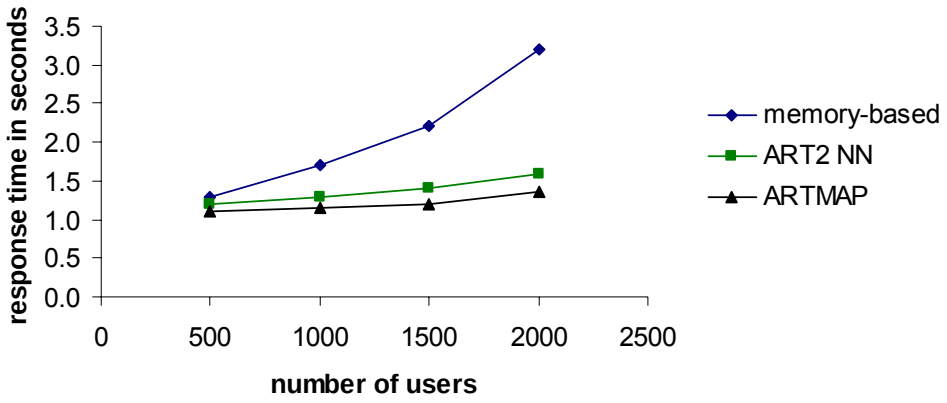


Figure 4. Response time.

Conclusion

Generally, the task in collaborative filtering is to predict the votes of a particular user from a database of user votes from a sample or population of other users. This paper presents a model based-approach to collaborative filtering by using supervised ARTMAP neural network (NN). Proposed algorithm is based on formation of reference vectors that make a CF system able to classify user profile patterns into classes of similar profiles, which forms the basis of a recommendation system. Experimental results presented here used the EachMovie data set. The first group of experiments shows classification accuracy in condition of growing database of votes. It can be seen the ARTMAP network provides better performance than both ART2 network and the popular memory-based neighborhood algorithm. The second group of experiments shows the advantage of the proposed ARTMAP model over both ART2 model and the memory-based method comparing response times in condition of growing number of users.

Bibliography

[1] Carpenter, G., S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Computer Vision, Graphics, and Image Processing, vol. 37, pp. 54-115, 1987.

[2] Carpenter, G., & Grossberg, S. ART 2: Stable self-organization of pattern recognition codes for analog input pat- terns. Applied Optics, 26, 4919-4930,1987.

[3] Carpenter, G., S. Grossberg, and J. H. Reynolds, ARTMAP: Supervised real-time learning and classification of non-stationary data by a self-organizing neural network, Neural Networks, vol. 4, pp. 565-588, 1991.

[4] Nachev, A., I. Ganchev, A Model-Based Approach to Collaborative Filtering by Neural Networks, In proceedings of the 2004 International Conference in Computer Science and Computer Engineering IC-AI'05, Las Vegas, 2005

[5] Goldberg, D., D. Nichols, B. Oki, D. Terry, Using collaborative filtering to weave an information tapestry, Communications of the ACM 35, 1992

[6] Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, In Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp. 175–186, 1994.

[7] Shardanand, U., P. Maes, Social information filtering: Algorithms for automating "word of mouth." In Proceedings of Computer Human Interaction, pp. 210–217, 1995.

[8] Breese, J., D. Heckerman and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, pp. 43–52, 1998.

[9] Basu, C. Hirsh, H. and Cohen, W. Recommendation as classification: Using social and content-based information in recommendation, In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), pp. 714–720., 1998

[10] Billsus, D., and M. Pazzani, Learning collaborative information filters, In Proceedings of the Fifteenth International Conference on Machine Learning, pp. 46–54, July 1998.

[11] Schafer, J., J. Konstan, and J. Riedl. Recommender systems in e-commerce. In Proceedings of the ACM Conference on Electronic Commerce (EC-99), pp. 158–166, 1999.

[12] Mc Jones, P., EachMovie collaborative filtering data set, DEC Systems Research Center, http://www.research.compaq.com/SRC/eachmovie/.

[13] Hawes, D. Information literacy in the business schools, In: Sep/Oct Journal of Education in Business, 70, pp 54-62, 1994

Author's Information

Anatoli Nachev – Information Systems, Dept. of A&F, NUI Galway, Ireland; e-mail: anatoli.nachev@nuigalway.ie

SYNTHESIS METHODS OF MULTIPLE-VALUED STRUCTURES OF LANGUAGE SYSTEMS

Mikhail Bondarenko, Grigorij Chetverikov, Alexandr Karpukhin, Svetlana Roshka, Zhanna Deyneko

***Abstract**: The basic construction concepts of many-valued intellectual systems, which are adequate to primal problems of person activity and using hybrid tools with many-valued of coding are considered. The many-valued intellectual systems being two-place, but simulating neuron processes of space toting which are different on a level of actions, inertial and threshold of properties of neurons diaphragms, and also modification of frequency of following of the transmitted messages are created. All enumerated properties and functions in point of fact are essential not only are discrete on time, but also many-valued.*

***Keywords:** intelligent system, hybrid logic, multiple-valued logic, multi-state element.*

***ACM Classification Keywords**: C.0 Computer Systems Organization: System architectures*

Introduction

The basic construction concepts of many-valued intellectual systems (MIS), which are adequate to primal problems of person activity and using hybrid tools with many-valued coding [1, 2] are considered. With materialism of a point of view these concepts are agreed with the dialectic laws opened by a man and their manifestations in problems connected with creation of identification systems prediction and recognition of imagery in which the interactive operational mode is a main part of the whole complex of intellectual properties.

Those are, for example, the law of unity and struggle of contrasts – as availability in parallel operating in space and time of mechanisms both discrete, and continuous mapping objects of plants; the law of transition from quantitative changes to qualitative-quantitative changes of gradation levels of brightness and the colors result in qualitative changes in mapping of objects; the law of negation of negation – as a changes and alternation of coding indications of messages about objects in neurons of a brain – from space to temporal and from two-place to many-valued [3,5].

In particular, in works the accent on the concept of neuro-physiologic and neuro-cybernetic aspects of alive brain mechanisms is made. It is connected with the following natural neuron structures from nervous cells – neurons,

essentially are highly effective recognizing systems and, for this reason, is of interest not only for doctors and physiologists, but also for the experts designing artificial intelligence systems. However direct transfer of research results of neuro-physiologists in engineering practice is now impossible because of a lack of an appropriate bioelectronic technology and an element basis, that has led to development and creation of a set of varieties of artificial neurons realized on the elements of the impulse technology.

But also here there were complications because of non-adequate neuron models to a set of the demands made of MIS. Creation of neuro-like models on the basis of multiprocessor in inputting systems technology with programmed architecture, in particular, on the basis of digital integrating structures is offered as the alternative in works [1–4]. Thus, retaining Neumann structure a MIS are created, being essentially two-place, but simulating neuron processes of space toting different on a level of actions, inertial and threshold properties of neuron diaphragms, as well as variation of recurrence frequency of transmitted messages. Though it is obvious that all enumerated properties and functions in point of fact, are, essential, not only discrete on time, but also many-valued (are discrete on a level).

As the corollary, non-adequacy of used principles of coding and element basis to simulated processes entails a redundancy, complication and non evidence of used mathematical and engineering means of transformations, loss of a micro level of parallelism in handling expected fast acting and flexibility of restructuring without essential modifications of architecture and connections.

Structurally Functional Cell Model of a Many-Valued Intellectual System

The originating complications [1], in creation of a many-valued intellectual system (MIS) promote moving out of the adequacy concept of many-valued logic and structures to of MIS creation problems with desirable properties and possibilities.

Therefore, for disclosure of use paths of a knowledge backlog in the field of many-valued coding and structures in MIS creation the conceptual structurally functional model of a MIS cell (Fig.1) is offered.

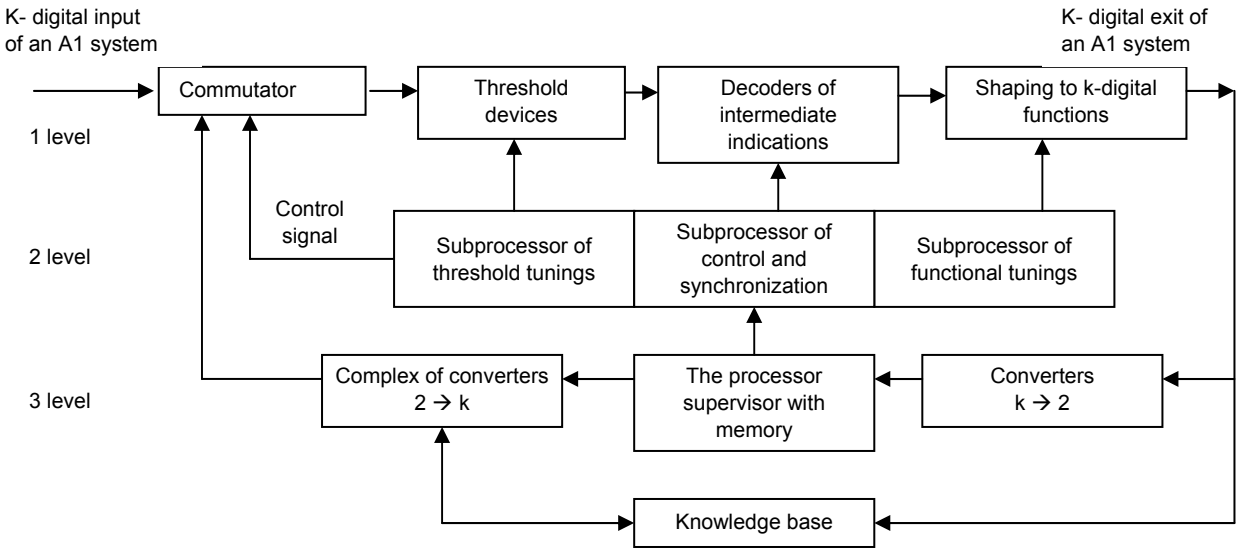


Fig. 1. A conceptual structurally functional model of a MIS cell.

Each MIS is characterized by a set of functions fulfilled by it by blocks, which realize functions and information interchanges. In accordance with solved problems, the structurally functional cell breaks up to three hierarchical levels: functional (analytic-synthetic) – level 1; tactical (analyses-coordination) – level 2; strategic (coordination) – level 3.

The MIS cell increases on a function level both on inputs, and on outputs, and it is integrated with other meshes on inputs of decoders of intermediate indications; at a tactical level – through the analyze-coordination processor;

at a strategic level – through the processor-supervisor and knowledge base. The conceptual model of a MIS cell is based on the concept of symbiosis of two- and many-valued tools of data processing, therefore at a strategic level it contain complexes of converters of the data representation form – converters from a two-place code to many-valued (2→K) and back (K→2). Obviously, that their use in MIS determines, at what level the problems, are solved in what logic and with what speed (what channel capacity of MIS). Besides the application of these tools excludes necessity of an operator work with two-place translators in input – output of data.

The new principle of the COMPUTER construction is offered, in which the principle of organization of brainwork simultaneously with a principle of programmed control assumes ass a basis. The principle of organization of brainwork assumes as a basis of operation of such COMPUTERS, in classical element basis it will be for more to Hilbert machines than for nowadays existing Neumann machines, the basis of which is the principle of programmed control realized rather slowly.

Formalization of Construction Principles of Many-Valued Spatial Structures

In the generalized form the two-input universal k-valued structure of a spatial type contains two recognition elements (RE), the control unit (CU), the matrix selector (MS), commutator (C), and keys (K) or the digital-to-analog converter (DAC) [2,3] (Fig.2).

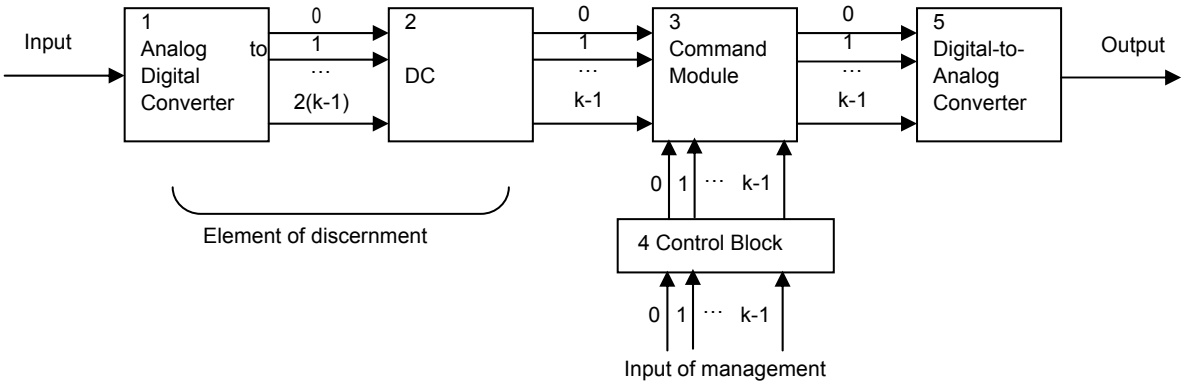


Fig. 2. Universal Multiple-Valued Functional Converter.

The logic of the decoders operation in recognition elements 1,2 is described by the following equation system:

$$f_0 = (x_0, x_{1,\dots,} x_{k-1}) = y^0,$$
$$f_1 = (x_0, x_{1,\dots,} x_{k-1}) = y^1,$$
……..,
$$f_{k-1} = (x_0, x_{1,\dots,} x_{k-1}) = y^{k-1}.$$

or in the explicit form at the algebra language of finite predicates [1]:

$$y_{1,2}^0 = \overline{x_1},$$
$$y_{1,2}^1 = x_1 \cup \overline{x_2},$$
$$y_{1,2}^2 = x_2 \cup \overline{x_3},$$
……,
$$y_{1,2}^{k-1} = x_{k-1}.$$

where x_i and $\bar{x}_i$ $(i=\overline{0,k-1})$ – signals of direct and inversion outputs of the ADC units in recognition elements 1,2. The logic of the matrix selector is described by the following equation system:

$b_{00}=y_1^0\cup y_2^0, b_{01}=y_1^0\cup y_2^1,...,b_{0(k-1)}=y_1^0\cup y_2^{k-1}\ b_{10}=y_1^1\cup y_2^0, b_{11}=y_1^1\cup y_2^1,...,b_{1(k-1)}=y_1^1\cup y_2^{k-1}$

……..

$b_{(k-1),0}=y_1^{k-1}\cup y_2^0;\ b_{(k-1),1}=y_1^{k-1}\cup y_2^1;\ \ldots\ b_{(k-1),(k-1)}=y_1^{k-1}\cup y_2^{k-1}$

where b_{ij} $(i,j=\overline{0,k-1})$ – output logical signals of the matrix selector4. The commutator has two groups by k inputs: the signals from the selector are applied to the first group and control signal values are, applied to the second group. In the explicit from the commutator operation is described by the following system:

$$b^{k_0}l^0\cup b^{k_0}l^1\cup\ldots\cup b^{k_0}l^{k-1}=z^{k_0},$$
$$b^{k_1}l^0\cup b^{k_1}l^1\cup\ldots\cup b^{k_1}l^{k-1}=z^{k_1},$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$b^{k_{k-1}}l^0\cup b^{k_{k-1}}l^1\cup\ldots\cup b^{k_{k-1}}l^{k-1}=z^{k_{k-1}}.$$

As all k of keys of the output shaper are constantly connected to corresponding k-values of output signals the function values selected by the commutator and the control unit, respectively, will arrive in the converter output (structure) in the course of variations of k-valued functions on the converter inputs. The process control of the logic recommutations is carried out under the action of external control signals.

Modeling and Realization

One of ways of realization of multiple-valued elements is the frequent-harmonic multi-stable element, which basis is the self-excited oscillator with a nonlinear resonant circuit, which is synchronized by an external voltage source.

At apparent simplicity, such circuit due to nonlinear properties has a lot of stable states. This circuit is supplied by a sequence of pulses with high period-to-pulse duration ratio. The control of circuit is carried out by feed of control pulses in a circuit of automatic bias. The process comes to an end then, when the resonant circuit appears tuned on next harmonic of a supplied voltage. Besides, the voltage of automatic bias changes too. Thus, the multi-state element has two attribute of each stable state – a voltage and frequency.

Using parabolic approximation of the characteristic of the transistor, we shall receive the following equation for a charge on nonlinear capacity of MOS-structure.

$$\frac{d^2\chi}{dt^2}+\omega^2\chi=-F_2\left(\frac{d^2\chi}{dt^2},\frac{d\chi}{dt},\chi\right)+S^*(t);$$

$$F_2\left(\frac{d^2\chi}{dt^2},\frac{d\chi}{dt},\chi\right)=\varepsilon\frac{d^2\chi}{dt^2}\left(\chi^2\alpha+\chi(1+\beta)+\gamma\right)+\varepsilon\left(\frac{d\chi}{dt}\right)^2\cdot\left(1+\chi(1+\beta)+\gamma\right)+$$
$$+\frac{d\chi}{dt}\left(h_1+\frac{\varepsilon}{\tau}(1+\chi^2\alpha+\chi(1+\beta)+\gamma\right)+\omega^2\left(1+\frac{\chi^3}{3}+\chi^2\alpha+\chi\beta+\gamma-B^*\right)$$

where χ – normalized charge on capacity of MOS-structure; ω – resonant frequency of a resonant circuit; ε – small parameter; $S^*(t)=N_kP(\tau_1)\sin\left[k\Omega,t+\gamma_k(\tau_1)\right]$

$$C_k=C_{k0}(1+b)^{\frac{1}{2}};\ s_0^*=\frac{s_1}{C_k};\ s_2^*=s_0^*+\frac{\xi}{\varepsilon};\ \lambda=s_1^*\varphi_k(1+b);$$

s_o, s_1 – coefficients of polynomial in approximation of the characteristic of the transistor;

φ_k – contact difference of potentials.

The solution of the this equation in the second approximation in case of the main resonance is

$$\chi = \alpha\cos\psi; \quad \psi = \nu t + \vartheta;$$
$$\frac{d\alpha}{dt} = \alpha\xi - \alpha^2\delta - \alpha\eta - N_k\frac{P(\tau_1)}{\omega+\nu}\cos[\gamma_k(\tau_1) - \vartheta]; \tag{1}$$
$$\frac{d\vartheta}{dt} = \omega - \nu + \chi + \frac{1}{\alpha}\xi + \alpha\theta + \alpha^2\sigma + N_k\frac{P(\tau_1)}{\alpha(\omega+\nu)}\sin[\gamma_k(\tau_1) - \vartheta],$$

where: ν – frequency of a synchronizing signal;

$$\xi = \frac{1}{\pi}\left(\left[h_1 + \frac{\varepsilon}{\tau_{\dot{y}}}(1+\gamma)\right]\left(\frac{\sin 2\psi_1}{4} - \frac{\psi_1}{2}\right) - \left(h_1 + \frac{\varepsilon}{\tau_e}\right)\left(\frac{\sin 2\psi_1}{4} + \frac{\pi}{2} - \frac{\psi_1}{2}\right)\right);$$
$$\theta = \frac{1}{\pi\omega}\left[H\left(\frac{\sin 3\psi_1}{12} + \frac{3}{4}\sin\psi_1\right) + \frac{1}{3}\varepsilon\omega^2(1+\gamma)\sin^3\psi_1\right];$$
$$H = \omega^2\alpha - \varepsilon\nu^2(1+\beta).$$

Let's consider conditions, at which the stable synchronous mode of stationary oscillations is possible. The values of amplitude and phase in a stationary mode are determined from system of the equations

$$R(a,\upsilon)=0 \; ; \; T(a,\upsilon)=0, \tag{2}$$

where $R(a,\upsilon), T(a,\upsilon)$ – right parts of the equations (1).

The conditions of stability of the solutions of the equations (2) are determined by the following inequalities:

$$\frac{da}{d\nu} \rangle\ 0 \quad \omega_e(a) \rangle\ \nu \ ;$$
$$\frac{da}{d\nu} \langle\ 0 \quad \omega_e(a) \langle\ \nu \ ,$$

where $\omega_e(a)$ – equivalent frequency of own oscillations.

The analysis of phase portraits of this dynamic system (frequency-harmonic multi-state element) has confirmed presence of stable modes in it.

By excluding a phase from the equations (2) it is possible to receive the equation for the amplitude-frequency characteristic [6].

The considered above frequency-harmonic multi-state element was realized as the hybrid thin-film integrated circuit with MOS-structure chip. Inductance elements were made as thin film LC structure [7]. The problem of it optimization [8] was solved.

The earlier received results get the importance in this time, when the semiconductor technology of manufacturing of the large scale integrated circuits for microprocessors practically has reached a physical limit of reduction of the size of components and width of interconnections. Alternative can be only use of artificial language systems, in which the elements of multiple-valued logic can be used. Experimental samples of the frequency-harmonic multi-state element were realized as thin-film integrated circuits in the standard case and can be used as elements of multiple-valued logic.

Conclusion

The problem solving of principles formalization of the structure organization of computing tools, thus ensures construction of the newest concept for systems of an artificial intelligence; application of space and temporal parallelism at structural and algorithmic levels; creation of procedural and functional languages, parallel machines

of knowledge bases and the inference. The problem solving of organization principles formalization of universal k-valued structures of a spatial type by tools of predicate and hybrid logic will ensure construction of a modern concept for artificial intelligence systems, application of spatial parallelism at structured and algorithmic levels; creation of functional languages of parallel machines of knowledge basis; application of symbiosis of two- and many-level heterogeneous coding.

One of circuit for realization of multiple-valued elements is the frequency-harmonic multi-state element which states are coding by amplitude and frequency. This element was made by thin film technology as hybrid integrated circuit.

Bibliography

[1] *M.F. Bondarenko, Z.D. Konopljanko, G.G. Chetverikov*. Osnovy teorii synteza nadshvydkodiuchikh structur movnykh sistem shtuchnogo intellectu, Monografia. – K.: IZMN, 1997. – 386 s.

[2] *M.F. Bondarenko, Z.D. Konopljanko, G.G. Chetverikov.* Osnovy teorii bagatoznachnikh struktur i koduvannya v sistemach shtuchnogo intelektu. – Kh.: Factor-Druk, 2003. – 336 s.

[3] *M.F. Bondarenko, S.V. Lyahovets, A.V. Karpukhin, G.G. Chetverikov*. Sintez shvidkodiuchikh structur lingvistichnich ob'ekhtiv., Proc. of the 9th International Conference KDS – 2001, St.Peterburg, Russia, 2001, s. 121–129.

[4] *M.F. Bondarenko, A.V. Karpukhin, G.G. Chetverikov.* Analiz problemi sozdaniya novich tekhnicheskich sredstv dlya realizazii lingvisticheskogo itterfeisa.Proc. of the 10th International Conference KDS – 2003, Varna, Bulgaria, June16–26, 2003, pp. 78-92.

[5] *M.F. Bondarenko, V.N. Bavykin, I.A. Revenchuk, G.G. Chetverikov.* Modeling of universal multiple-valued structures of artificial intelligence systems, Proc. of the 6th International Workshop "MIXDES'99", Krakow, Poland, 17-19 June 1999, pp. 131–133.

[6] *M.F. Bondarenko, A.V. Karpuhin, G.G. Chetveriko, Zh.V. Deyneko.* Application of a numerically – analytical method for simulation of non-linear resonant circuits.10 th International Conference Mixed Design Of Integrated Circuits And System (MIXDES 2003), Lodz, Poland, 26–28 June 2003,

[7] *V.V. Aleksandrov, A.V. Karpukhin*. Osobennocti konstruktivnogo rascheta i technologii izgotovlenija mikroelektronnich ustrojstv obmena informaciej. Izvestija visshich uchebnich zavedenij.Priborostroenije. Leningradskij institut tochnojj mechaniki i optiki. Tom. XX, N 5, 1977. – pp.120-124.

[8] *G.I. Yalovega., Yu.Kh. Loza*, *A.V. Karpukhin.* Matematicheskoe modelirovanie i optimizaciya mnogofunkzionalnich resonansnikh cepei. 26. Internationales Wissenschaftliches Kolloquim.Technishe Hochschule Ilmenau, 1981, Heft 2, Vortragsreihen A3, A1.

Authors' Information

Mikhail Fedorovich Bondarenko – Rector, Prof., State National University of Radio-Electronics P.O. Box 14, Lenin's avenue, Kharkov, 61166, Ukraine.

Grigorij Grigorjevich Chetverikov – c.t.s. State National University of Radio-Electronics P.O. Box 14, Lenin's avenue, Kharkov, 61166, Ukraine.

Alexandr Vladimirovich Karpukhin – c.t.s., State National University of Radio-Electronics P.O. Box 14, Lenin's avenue, Kharkov, 61166, Ukraine; e-mail: kav@kture.kharkov.ua

Svetlana Alexandrovna Roshka – Ph.D. student, State National University of Radio-Electronics P.O. Box: 14, Lenin's avenue, Kharkov, 61166, Ukraine.

Zhanna Valentinovna Deyneko – Ph.D. student, State National University of Radio-Electronics P.O. Box: 14, Lenin's avenue, Kharkov, 61166, Ukraine.

SIGNAL PROCESSING UNDER ACTIVE MONITORING

Oleksii Mostovyi

Abstract: *This paper describes a method of signal preprocessing under active monitoring. Suppose we want to solve the inverse problem of getting the response of a medium to one powerful signal, which is equivalent to obtaining the transmission function of the medium, but do not have an opportunity to conduct such an experiment (it might be too expensive or harmful for the environment). Practically the problem can be reduced to obtaining the transmission function of the medium. In this case we can conduct a series of experiments of relatively low power and superpose the response signals. However, this method is conjugated with considerable loss of information (especially in the high frequency domain) due to fluctuations of the phase, the frequency and the starting time of each individual experiment. The preprocessing technique presented in this paper allows us to substantially restore the response of the medium and consequently to find a better estimate for the transmission function. This technique is based on expanding the initial signal into the system of orthogonal functions.*

Keywords: *mathematical modelling, active monitoring, frequency and phase fluctuation.*

ACM Classification Keywords: *I.6.1 Simulation Theory.*

Introduction

In June 2001 in Kiev, Ukraine, by Kiev Institute of Geophysics there was conducted an experiment to determine the properties of the monument "Ukraine" such as eigenfrequencies and damping decrement. In particular the responses of the monument to a series of low power ambient perturbation were measured [1-4]. This paper describes a method of preprocessing of the observed signal that allows to substantially restore the response of the medium and to find the properties of the monument more precisely.

The Mathematical Model of Active Monitoring with Frequency and Phase Fluctuations of Sound Signal

For the i-th experiment in a series of M experiments the model of active monitoring can be represented as follows:

$$y_i(t) = S\left(t, \tau_i, \vec{h}_i\right) * H(t) + n_i(t), \quad t \in \left(\tau_i, \tau_i + T\right) \tag{1}$$

where $y_i(t)$ is the response of the medium to the exploring signal $S\left(t, \tau_i, \vec{h}_i\right)$, which depends on the vector of parameters $\vec{h}_i$ in the i-th experiment. This signal is convoluted with the response of the medium $H(t)$ to the delta-function signal $\delta(t)$. Also, $n_i(t)$ is the additive background noise, which accompanies the experiment, T is the duration of one experiment, $\left(\tau_i, \tau_i + T\right)$ is the time interval of the i-th experiment realization, and $*$ is the symbol of the convolution operator. The experiment is created in such a manner that the energy $\mathrm{E}\left[S\left(t, \tau_i, \vec{h}_i\right) * H(t)\right]$ of the registered by sensors signal and the energy $\mathrm{E}\left[n_i(t)\right]$ of the natural background are commensurable in the selected metrics, i.e. the influence of the experiment on the environment is negligible. The model takes into account that nonlinear effects in the experiment might be neglected. Therefore the linear procedure for the interaction of the medium and the exploring signal in the form of convolution is selected.

The next model defines the full experiment:

$$y(t) = \sum_{i=1}^{M} y_i(t) \tag{2}$$

The Probing Signal Model

Consider a signal $S\left(t, \tau_i, \vec{h}_i\right)$ that depends on the vector of parameters $\vec{h}_i = \left\{h_{i1}, \ldots, h_{i\,N+2}\right\}$, first N components of which $h_{i1}, \ldots, h_{i\,N}$ are included into the model linearly. These parameters determine the form of the exploring signal in the i-th experiment, τ_i is the starting time of the i-th experiment, $h_{i\,N+1}$ is the fluctuating frequency $\omega_{0\,i}$, and $h_{i\,N+2}$ is the fluctuating phase ψ_i.

One considers the signal to be a stationary, physically realizable wave, i.e. it satisfies the following conditions:

$$\begin{aligned}
&\int_0^{\infty} S\left(t, \tau_i, \psi_i, \vec{h}_i\right) dt = 0 \text{, for} \forall\ h_i; \\
&S\left(t, \tau_i, \psi_i, \vec{h}_i\right) = \begin{cases} S\left(t - \tau_i - \psi_i, h_{1\,i}, \ldots, h_{i\,N+2}\right), & \text{for } t \geq \tau_i \\ 0, & \text{for } t < \tau_i. \end{cases}; \\
&\int_0^{\infty} \left(S\left(t, \tau_i, \psi_i, \vec{h}_i\right)\right)^2 dt < \infty. \\
&\int_0^{\infty} S\left(t, \vec{h}_i\right) dt = 0 \text{, for } \forall\ h_i.
\end{aligned} \tag{3}$$

The last equality in (3) means that the signal is a wave, i.e. the signal does not leave after-effects and consequently it does not change the constant constituent of the medium.

The same conditions hold for the response of the medium $H\left(t\right)$.

The latter circumstance allows us to determine the duration of one experiment T (taking into account statistical characteristics of the background noise). For this purpose the noise level ε is required to be much less or at least less than the natural background energy level. Here

$$\varepsilon = \sqrt{\int_T^{\infty} \left(S\left(t, \tau_i, \psi_i \vec{h}_i\right)\right)^2 dt}$$

Within some tolerance one can represent the signal as a linear combination of the truncated orthogonal basis functions (in some metric) on the interval of length T:

$$\varphi_k(t - \tau_i, k\omega_{0\,i})\chi\left(t, \tau_i, \tau_i + T\right);\ \omega_{0\,i} = \omega_0 + \Delta\omega_i;\ T = \omega_0 \pi.$$

$$S\left(t, \tau_i, \psi_i, \vec{h}_i\right) = \sum_{k=1}^{N} h_{i\,k} \varphi_k(t - \tau_i - \psi_i, k\omega_{0\,i})\chi\left(t, \tau_i, \tau_i + T\right) \tag{4}$$

Here ψ_i is the fluctuating phase with zero expectation; $\omega_{0\,i} = \omega_0 + \Delta\omega_i$ is the random frequency with mean ω_0.

The set $\vec{\omega} = \left\{\omega_1, \omega_2, \ldots, \omega_M\right\}$ is considered to be the set of outcomes of the random variable $\omega_i = \omega_0 + \Delta\omega_i$ fluctuating around ω_0, here fluctuations carry out on the closed interval $\left[\omega_1^*, \omega_2^*\right]$ and

$\left|\omega_1^* - \omega_2^*\right| < \omega_0$. In our case we can approximate the partial density of the parameter *h* distributed on the interval $\Delta = [h_1, h_1 + \varepsilon]$ of length ε with a beta distribution with parameters $\gamma, \eta, \varepsilon$. Varying these parameters one can obtain different forms of approximation of the density (5), which has the following form [5]:

$$e(h,\gamma,\eta,\varepsilon) = \begin{cases} \dfrac{1}{\varepsilon}\dfrac{\Gamma(\gamma+\eta)}{\Gamma(\gamma)\Gamma(\eta)}\left(\dfrac{h}{\varepsilon}-h_1\right)^{\gamma-1}\left(1-\dfrac{h}{\varepsilon}-h_1\right)^{\eta-1}; h \in \Delta; \\ 0, \text{when} h \notin \Delta; \quad 0<\gamma, 0<\eta; \quad \Delta = \left((k-1)T, (k-1)T+\varepsilon\right) \end{cases} \tag{5}$$

The particular case of the beta distribution with the parameters $\gamma = \eta = 1$ is the uniform density. Let us pay especial attention to the uniform density since it has the maximum entropy among all the distributions on the closed interval.

Here entropy $H(\alpha)$ is a measure of uncertainty of an experiment α, in the result of which the events $A_1, A_2, \ldots, A_n$ with corresponding probabilities $p_1, p_2, \ldots, p_3$ occur. The (Shannon) entropy $H(\alpha)$ is defined as $H(\alpha) = -\sum_{i=1}^{n} p_i \log_2(p_i)$ [6].

In our case as a result of the *i*-th experiment the parameter of the probing signal appears in the *i*-th sub-region of the set (divided with the points $x_1, x_2, \ldots, x_n$) of all possible values of this parameter. We consider the probability of this event to be defined as a beta distribution with the parameters η, γ:

$$p_i = \frac{\Gamma(\eta+\gamma)}{\Gamma(\eta)\Gamma(\gamma)} \int_{x_{i-1}}^{x_i} x^{\eta-1}(1-x)^{\gamma-1}\,dx, \quad \eta > 0,\ \gamma > 0. \tag{6}$$

Entropy becomes the function of the parameters η, γ and the number n of intervals of the partition. Entropy is represented on figure 1, with $0.1 \le \eta \le 10$, $0.1 \le \gamma \le 10$ and $n = 20$.

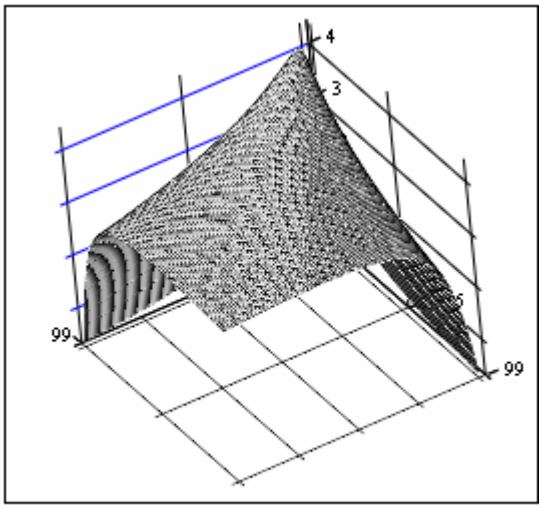


Figure 1. Entropy of the beta distribution as a function of parameters $0.1 \le \eta \le 10$ *and* $0.1 \le \gamma \le 10$. Here $n = 20$.

We can see on figure 1 that the entropy is a uni-modal surface with the maximum at $\eta = \gamma = 1$, i.e. the maximum is reached when the distribution on the given interval is uniform.

As a result if we assume that the distribution of the parameters of the signal is uniform, the minimum information about the experiment is introduced and the estimate of the parameters will be the worst of all possible ones for such a length of the interval. This means that using any other a priory distribution we will only improve the estimate.

The same argument holds for the fluctuating phase $\vec{\psi} = \{\psi_i\};\ i = 1, \ldots, M$.

These parameters define the truncated system of basis functions:

$$\vec{\varphi}_i(t, \tau, \psi_i, \omega_{0i}) = \left\{\varphi_k(t - \tau - \psi_i, \omega_{0i}k)\chi\left(t, \tau_i, \tau_i + T\right)\right\},\ k = 1, \ldots, N \tag{5}$$

The parameters T, ω_{0i}, ψ_i are included into the model non-linearly.

Here $\chi\left(t, \tau_i, \tau_i + T\right) = \begin{cases} 1, & t \in \left(\tau_i, \tau_i + T\right), \\ 0, & t \notin \left(\tau_i, \tau_i + T\right); \end{cases}$ - is the characteristic function of the interval.

Let's consider the scenario when the values $\tau_i = \left(i - 1\right)T$ are deterministic.

We consider the case, when the set of vectors $\vec{h}_1, \ldots, \vec{h}_M$, and values ω_{0i}, ψ_i is sampled from the set of the possible values of the vector $\vec{h}, \omega_0, \psi$ with an a priori known distribution $P\left(\vec{h}, \omega_0, \psi\right)$ and the determined values $\tau_i = \left(i - 1\right)T$. I.e. the stochastic parameters $\vec{h}, \omega_0, \psi$ and the stochastic additive value $n\left(t\right)$ characterize the stochastic nature of the process $y\left(t\right) = \sum_{i=1}^{M} y_i\left(t\right)$. The information about the medium is included into the process via the deterministic function $H\left(t\right)$, the response of the medium to the delta function.

The response of the medium in the i-th experiment $y_i(t)$ is described by the following equation:

$$y_i\left(t\right) = \left(\sum_{k=1}^{N} h_{ik}\left(\int_{\tau_i}^{\tau_i + T} \varphi_k\left(t - \tau - \psi_i, \omega_{0i}k\right)H(\tau)d\tau\right)\right)\chi\left(t, \tau_i, \tau_i + T\right) + n_i\left(t\right). \tag{7}$$

Thus the result of the series of M experiments $y\left(t\right)$ is:

$$\begin{aligned} y\left(t\right) = \sum_{i=1}^{M} y_i\left(t\right) = \sum_{i=1}^{M}\left(\sum_{k=1}^{N} h_{ik}\left(\int_{\tau_i}^{\tau_i + T} \varphi_k\left(t - \tau - \psi_i, \omega_i k\right)H\left(\tau\right)d\tau\right)\right) \\ \times\chi\left(t, \tau_i, \tau_i + T\right) + n_i\left(t\right),\quad t \in \left(0, MT\right) \end{aligned} \tag{8}$$

Data-processing

Let's propose the following model for the processing procedure of the observed data:

$$\frac{1}{I}\sum_{i=1}^{M} y_i\left(t-(i-1)T,\vec{h}_i\right)+$$
$$+\frac{1}{I}\sum_{i=1}^{M} n\left(t-(i-1)T\right)=\hat{E}\left[y(t)+n(t)\right]=\hat{E}\left[y(t)\right]+\hat{E}\left[n(t)\right];\ I=MT;\ t\in(0,T) \tag{9}$$

Here $\hat{E}\left[y(t)+n(t)\right]$ is the expectation estimator of the response of the medium and additive background noise, and $I=MT$ is the total time of monitoring.

Suppose $\hat{E}\left[n(t)\right]=0$, then the procedure of data processing reduces to the calculation of $\hat{E}\left[y(t)\right]$.

On one hand we have equation (9), on the other hand we can formally calculate the average of the superposition of the responses of the medium and additive background noise using the following formula (taking into account the fact that $\vec{h},\psi$ and $\Delta\omega$ are mutually independent):

$$E\left[y(t)\right]=\int_{R_{\vec{h}}}\int_{R_\psi}\int_{R_{\Delta\omega}}\left(\sum_{k=1}^{N}h_k\left(\int_0^T\varphi_k(t-\theta-\psi,(\omega_0+\Delta\omega)k)H(\theta)d\theta\right)\right)\chi\left(t,\tau,\tau+T\right)$$
$$\times dP(\vec{h})dP(\psi)dP(\Delta\omega)+E\left[n(t)\right]$$
$$=\sum_{k=1}^{N}\int_0^T H(\theta)\int_{R_{\vec{h}}}h_k\int_{R_\psi}\int_{R_{\Delta\omega}}\varphi_k(t-\theta-\psi,(\omega_0+\Delta\omega)k)\chi\left(t,\tau,\tau+T\right) \tag{10}$$
$$\times dP(\vec{h})dP(\psi)dP(\Delta\omega)d\theta$$

Let's denote:

$$\tilde{\varphi}_k(t-\theta)=\int_{R_h}\int_{R_\psi}\int_{R_{\Delta\omega}}\varphi_k(t-\theta-\psi,(\omega_0+\Delta\omega)k)\chi\left(t,\tau,\tau+T\right)dP(\vec{h})dP(\psi)dP(\Delta\omega) \tag{11}$$

and

$$\hat{h}_k=\int_{R_{\vec{h}}}h_k dP\left(\vec{h}\right). \tag{12}$$

Assuming independence of the fluctuating parameters, equation (9) takes the following form:

$$E\left[y(t)\right]=\sum_{k=1}^{N}h_k\int_0^T H(\theta)\tilde{\varphi}_k(t-\theta)d\theta \tag{13}$$

The ultimate purpose of the experiment was to evaluate the response of the medium to one powerful signal. This signal exceeds the noise level by many times. Depending on how much do we want the signal to exceed the noise, the number of the (individual) experiments M is chosen for accrual of the signal. However the procedure of accumulation leads to considerable distortions especially in the high-frequency domain. One can eliminate these distortions by solving equation (13) with respect to $H(t)$. The calculation of the vector of functions $\tilde{\varphi}(t)=\left\{\tilde{\varphi}_k(t)\right\},k=\overline{1,N}$ is not difficult, since one can always receive a priori distributions of the parameters of the generated probing signals, and we can plug in the estimator $\hat{E}\left[y(t)\right]$ to the left hand side of equation (13).

Thus the problem is reduced to finding the solution of integral equation (13). One might look for the solution of the form:

$$H(t) = \sum_{q=1}^{Q} H_q \phi_q(t) \tag{14}$$

then

$$\hat{E}\left[y(t)\right] = \sum_{k=1}^{N} \hat{h}_k \sum_{q=1}^{Q} H_q \int_0^T \phi_q(\theta)\tilde{\varphi}_k(t-\theta)d\theta = \sum_{k=1}^{N} \hat{h}_k \sum_{q=1}^{Q} H_q \Psi_{k\,q}(t). \tag{15}$$

Here

$$\Psi_{k\,q}(t) = \int_0^T \phi_q(\theta)\tilde{\varphi}_k(t-\theta)d\theta. \tag{16}$$

Solving the latter problem with respect to $\vec{H}$ we obtain the desirable result.

The Analysis of Illustrative Examples

Solving the fragment of the direct problem (11) for one harmonic allows us to investigate the evolution of the signal during the accumulation resulted from its frequency and phase fluctuations. For the harmonic with number *k* the analytical solution of the direct problem for the uniform distribution of frequency fluctuations (the case, which has the maximum entropy) is given by the following expression:

$$\varphi_k(t) = \frac{-2\sqrt{2T}\sin\left[\left(\omega_0 + \frac{\varepsilon_2+\varepsilon_1}{2}\right)k\frac{t}{T}\right]\sin\left[\frac{\varepsilon_2-\varepsilon_1}{2T}kt\right]}{k(\varepsilon_2-\varepsilon_1)t}\chi(t,0,T). \tag{17}$$

From expression (17) we see that as the signal accumulates the harmonic is superimposed with the branch of the hyperbola whose coefficient equals to the number of the harmonic $\frac{1}{k(\varepsilon_2-\varepsilon_1)t}$, and the frequency of beating $\sin\left[\frac{\varepsilon_2-\varepsilon_1}{2T}kt\right]$ that is proportional to the number of the harmonic and the length of the time interval (on which the fluctuations of frequency are distributed). Figure 2 shows the corresponding harmonic number 31, which is undistorted (light line) and distorted (bold line) as the result of accumulation of the signal. Here $\omega_0 = \pi$ $\varepsilon_1 = 0$, and $\varepsilon_2 = 0.1$.

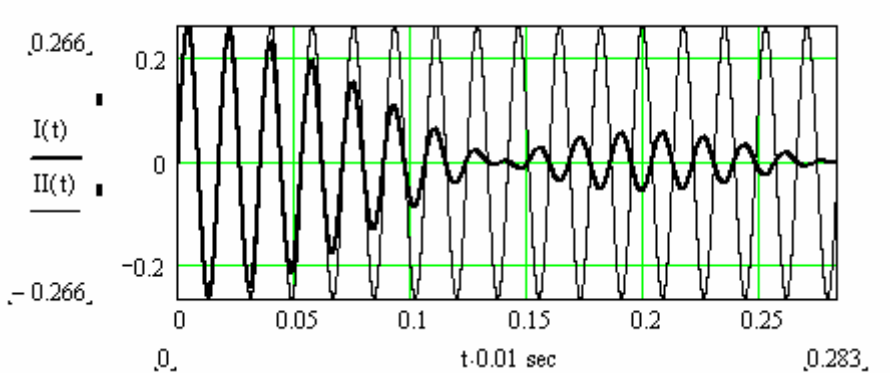


Figure 2. Undistorted harmonic (light line II(t)) and distorted harmonic (bold line I(t)) as the result of accumulation of the signal. Here $\omega_0 = \pi$ $\varepsilon_1 = 0$, *and* $\varepsilon_2 = 0.1$. *The abscissa axis represents time in seconds; the ordinate axis corresponds to the amplitude of the signals in relative units.*

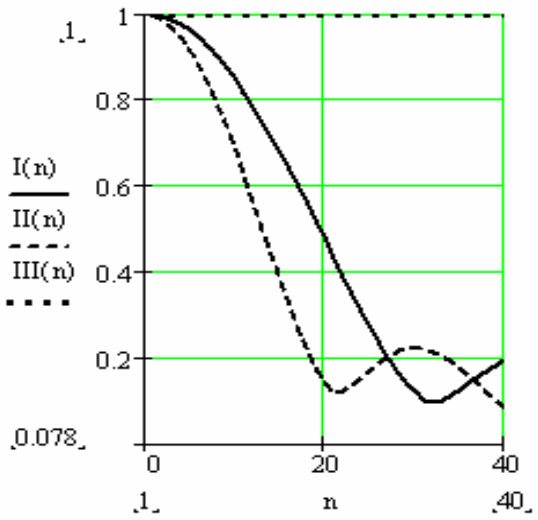


Figure 3. Change of the norm of the harmonic as a function of its number resulting from the accumulation of signals. I(n) and II(n) correspond to different values of the entropy H of the a priory distribution of the fluctuating frequency.

I(n) – n=200, $n = 200, \Delta = 1, H = 2.301$

II(n) - $n = 200, \Delta = 0.3, H = 1.778$

III(n) – norms of the undisturbed harmonics. Here Δ *is the interval of the fluctuating frequency.*

Example

Let us consider an example of the signal restoration procedure for a simple oscillator with a dumping term. Here the signal is a sinusoid of some frequency that is modulated with a decreasing exponent. The frequency of sinusoid is fluctuating.

Being simple enough the signal clearly reflects the essence of the processes. In a linear approximation one can describe most of the surrounding us objects with a system of the signals of this type. The Fourier's, Laplace's, Heaviside's images of the solution are well known, so the reader can imagine such a signal. In practice more complicated signals are used (the signal can be a function of time, frequency, phase or a function of all arguments simultaneously). The only difference of the considered example from solution (18) is that the signal has the finite length due to the presence of the characteristic function of the interval.

$$S(t, \vec{h}_i) = \theta_i \exp\{-\alpha_i t\} \sin\{\omega_i (t - \tau_i)\} \chi(t, \tau_i - \psi_i, \tau_i + T) \tag{18}$$

In (18) ψ_i is a phase shift in the i-th experiment.

In this case the vector of free parameters of the model, determining the signal, is $\vec{h}_i = \{h_{ik}\} = \{\tau_i, \theta_i, \alpha_i, \omega_i, \psi_i, T\}$, $k = 0, \ldots, 5$. It has only six components, second of which h_{i1} comes into the model linearly.

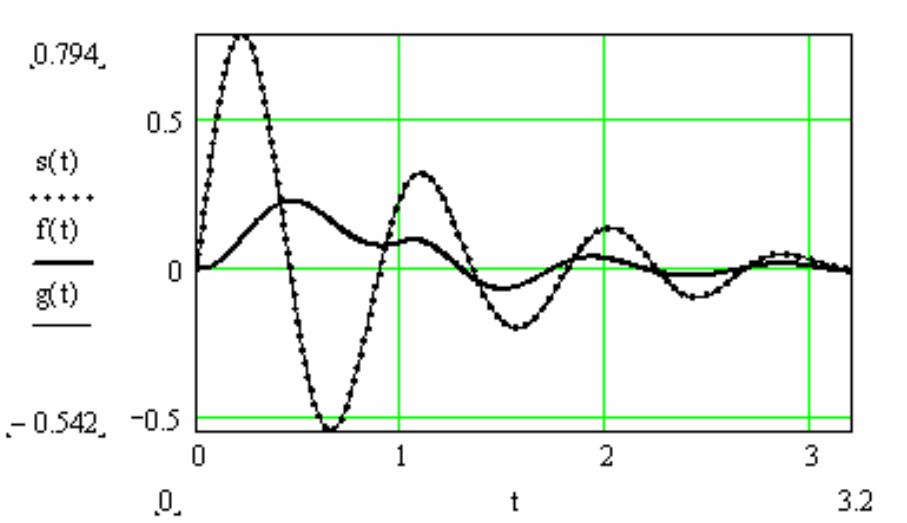


Figure 4. There are three curves on this picture: the first one, which is named $s(t)$, is the model of the sounding signal. The second one $f(t)$ is the signal misshaped by random frequency and phase fluctuations, the third one $g(t)$ is the reconstructed signal. The procedure of the signal reconstruction allows us to get the curve shape very close to the shape of the original one $s(t)$.

Conclusion

The results of the offered analysis of the procedure of active monitoring demonstrate considerable loss of information during accumulation of the medium response to the probing signal. It is important to emphasize, that the higher the signal frequency is, the greater the corresponding loss of information becomes. The preprocessing technique proposed in this paper allows us to conduct the correction of the experimental results and substantially restore the response of the medium.

Bibliography

1. Gui A. et al. Model and Experimental Studies of the Identification of Oil/Gas Deposits, Using Dynamic Parameters of Active Seismic Monitoring, Geophys. J., 2001, Vol. 20, pp. 895-9009.
2. Gui A. et al. Model of Active Structural Monitoring and decision-making for Dynamic Identification of buildings, monuments and engineering facilities. KDS 2003, Varna 2003, p. 97 – 102.
3. Kondra M. et al. Modern approaches to assurance of dynamic stability of the pillar type monument with an application of the wind tunnel assisted research and the site measuring of the dynamic characteristics. Eurodyn 2002, Swets & Zeitlinger, Lisse, 2002, p. 1511 - 1515.
4. Lebedich I. et al. Comprehensive aerodynamic and dynamic study of independence of Ukraine monument. Proceedings of the national Aviation University. 2' 2003, pp. 100 - 104.
5. Hahn Gerrald J. & Shapiro Samuel S. Statistical models in engineering. John Willey & Sons, Inc. New York. – London – Sydney. 1967. p 395.
6. http://mathworld.wolfram.com/search/entropy

Author's Information

Mostovyi Oleksii - Bank of America. New York, USA. e-mail: osm201@nyu.edu

ANALYSIS AND OPTIMIZATION OF SYNTHETIC APERTURE ULTRASOUND IMAGING USING THE EFFECTIVE APERTURE APPROACH

Milen Nikolov, Vera Behar

***Abstract**: An effective aperture approach is used as a tool for analysis and parameter optimization of mostly known ultrasound imaging systems - phased array systems, compounding systems and synthetic aperture imaging systems. Both characteristics of an imaging system, the effective aperture function and the corresponding two-way radiation pattern, provide information about two of the most important parameters of images produced by an ultrasound system - lateral resolution and contrast. Therefore, in the design, optimization of the effective aperture function leads to optimal choice of such parameters of an imaging systems that influence on lateral resolution and contrast of images produced by this imaging system. It is shown that the effective aperture approach can be used for optimization of a sparse synthetic transmit aperture (STA) imaging system. A new two-stage algorithm is proposed for optimization of both the positions of the transmitted elements and the weights of the receive elements. The proposed system employs a 64-element array with only four active elements used during transmit. The numerical results show that Hamming apodization gives the best compromise between the contrast of images and the lateral resolution.*

***Keywords**: Ultrasound imaging, Synthetic aperture, stochastic optimization.*

***ACM Classification Keywords**: J.3 Life and Medical Sciences: Medical information systems; I.5.4 Pattern recognition: Applications --- Signal processing; G.1.6 Numerical Analysis: Optimization --- Simulated annealing*

1. Introduction

Medical ultrasound imaging is a technique that has become much more prevalent than other medical imaging techniques since this technique is more accessible, less expensive, safe, simpler to use and produces images in real-time. However, images produced by an ultrasound imaging system, must be of sufficient quality to provide accurate clinical interpretation. The most commonly used image quality measures are spatial resolution, image contrast and frame rate. The first two image quality measures (resolution and contrast) can be determined in terms of beam characteristics of an imaging system beam width and side lobe level. In the design of an imaging system, the optimal set of system parameters is usually found as a tradeoff between the lowest sidelobe peak and the narrowest beam width of an imaging system. In a conventional ultrasound imaging system, the transducer is a phased array with a great number of elements (PA imaging systems). The quality of images produced by a PA system directly depends on the number of active channels used both in transmission and receiving. Thus, the conventional high-resolution PA imaging systems produce images at relatively high cost [1].

Conventional phased array imaging systems employ all elements of the transducer during both transmit and receive during each excitation cycle, while employing delays in order to steer the beam and scan a 2D plane. In receive mode, dynamic (or composite) focus is used, by adjusting the delays of transducer elements as a function of the depth being imaged. In transmit mode, usually the focus point is set in the middle of the region being imaged. At the focus point, the lateral beamwidth is the smallest (and the best lateral resolution is obtained there), while away from the focus point, the lateral beamwidth increases. The spatial resolution of the ultrasound image can be improved by using several transmit beams during the interrogation of each sector, each of which is focused at a different depth. It is done in modern ultrasound imaging systems at the cost of decrease of the frame rate, proportionally to the number of transmit foci [2]. An alternative way to obtain an appropriate spatial resolution, without the decrease of the frame rate, is to use the synthetic aperture technique. This method makes it possible to generate images with dynamic focusing, during both transmit and receive, while maintaining or even drastically decreasing the time of image acquisition.

In a classical Synthetic Aperture Focusing Technique (SAFT), only a single array element transmits and receives at each time. All the elements are excited sequentially one after the other, and the echoes received are recorded

and stored in computer memory. It reduces the system complexity and the frame rate, but requires data memory for all data recordings [3]. The main disadvantage of SAFT is the low signal-to-noise ratio (SNR) and as a result, the poor contrast resolution. In a Multi-element Synthetic Aperture Focusing (MSAF) method, at each time a group of elements transmits and receives signals simultaneously [4]. The transmitted beam is defocused to emulate a spherical wave. The SNR is increased compared to SAFT, in which only a single element is used in transmit and receive. In a Synthetic Transmit Aperture (STA) method, at each time one array element transmits a pulse, and all elements receive the echo signals [5]. Compared to conventional phased array imaging, the advantage of this approach is that a full dynamic focusing can be applied to the transmission and the receiving, producing the highest quality of images at the increased frame rate. The shortcoming is that a huge data memory is required for data recordings. For an *N*-element array, *N* echo recordings are required to form a conventional phased array image, and, however, $N \cdot N$ echo recordings are required to synthesize a STA image. This disadvantage can be overcame to some extent, if only a few elements, *M*, act as transmitters. In that case $M \cdot N$ echo recordings are required to synthesize a STA image, where $M<N$ [6]. This is equivalent to using of a sparse array in transmit. The sparse STA imaging acquires images at higher frame rates, which makes this method very attractive for real-time 3D-ultrasound imaging.

The relation between the employed effective aperture function and the resultant radiation pattern of the imaging system can be used as a strategy for analysis and for optimisation of an imaging system [7]. Since the two-way radiation pattern of a system is the Fourier transform of the effective aperture function, the transmitted and receiving radiation patterns can be optimised by selecting the appropriate transmit and receive aperture functions, to produce the "desired" effective aperture of the imaging system. Thus, when the desired effective aperture of a system is defined, it also provides the two-way radiation pattern that should be used, with the appropriate width of the main-lobe and its sidelobes. In synthetic aperture imaging, the transmit aperture function depends not only on the number of transmit elements, but also on their geometrical locations within the array (sparse synthetic aperture imaging). The received aperture function depends on the length of a physical array and the apodization weights applied to the receiver elements. Thus, the shape of the effective aperture function and, therefore, the shape of the two, one-way radiation patterns of a system, can be optimised depending on the positions of the element in transmit and the weights of the element in receive.

In this paper, it is shown how the effective aperture approach can be used for analysis and parameter optimisation of an ultrasound STA imaging system. Using this approach, the optimal set of system parameters (number of array elements, their configuration within an array) can be determined in result of a compromise between the lowest sidelobe peak and the narrowest beam width of the two-way radiation pattern of an imaging system. The comparison analysis of 3 types of imaging systems is done calculating their effective aperture function and the corresponding two-way radiation pattern using the computational environment of Matlab.

2. The Effective Aperture Concept

The effective aperture of an array represents an equivalent aperture that would produce identical two-way radiation pattern if the transmit aperture was a point source. An expression for the effective aperture of an array can be derived from a calculation of the two - way radiation pattern. Consider an uniformly spaced linear array of *N* elements with weighting $w(m)$ (m = 0, . . ., N . 1). The one- way far field beam pattern is

$$W(\theta) = \sum_{m=0}^{N-1} w(m) e^{-k^0 m d \sin(\theta)} \tag{1}$$

where d and k^o are the inter-element spacing and the wave number, respectively. This equation can also be described as a discrete Fourier transform (DFT) of the aperture function:

$$W(k) = DFT[w(m)] = \sum_{m=0}^{N-1} w(m) e^{-j\frac{2\pi}{N}km}, \; k=0, 1, \ldots, N-1 \tag{2}$$

in which the frequency index k maps into the beam angle θ by $\sin\theta = k\lambda/(Nd)$ where λ is the wavelength.

Since the round-trip beam pattern is the product of the transmit and receive beams

$$W_{RT}(\theta) = W_R(\theta) W_T(\theta) \tag{3}$$

using the DFT property, we get

$$W_{RT}(k) = DFT\left[w_R \otimes w_T\right] \tag{4}$$

where $\otimes$ denotes convolution and w_R and w_T are the apodization functions applied to the array elements in transmit and receive, respectively. Using (4), the effective aperture function of an imaging system is defined as

$$e_{RT} = w_R \otimes w_T \quad \text{and} \quad W_{RT}(\theta) = FFT(e_{RT}) \tag{5}$$

Thus, the round trip beam pattern is determined by the transmitted and the receiving aperture weightings. Every physical beam can be realized by forming the appropriate effective aperture.

3. Synthetic Aperture Imaging

First, the concept of synthetic aperture was originally used in radar for highly resolution imaging terrain, but it can be successfully used in ultrasound imaging systems as well. In this case, the benefit of the synthetic aperture is the reduction of system complexity and cost. Several methods were proposed to form a synthetic aperture for ultrasonic imaging. In SAFT imaging, at each time only a single array element transmits a pulse and receives the echo signal. (Fig. 1). The system complexity is reduced, because only a single set of circuit for transmit and receive is needed. In this case the effective aperture can be calculated by

$$e_N = \sum_{m=1}^{N} w_R(m) \otimes w_T(m), \text{ where } w_R(m) = w_T(m) = [0,0,\ldots,i_m,\ldots,0] \text{ and } i_m = 1 \tag{6}$$

In MSAF imaging, a group of elements transmit and receive signals simultaneously, and transmit beam is defocused to emulate a single element response (Fig. 2). The acoustic power and the signal-to-noise ratio are increased compared to SAFT where a single element is used. This method requires also memory for data recordings. In MSAF, a Kt-element transmit subaperture sends an ultrasound pulse and echo signals are recorded at a Kr-element receive subaperture. At the next step, one element is dropped and a new element is included to the transmitted and receiving subaperture, repeating the transmission and receiving process. Usually K_t=K_r=k. The effective aperture is:

$$e_N = \sum_{m=1}^{N-k+1} w_R(m) \otimes w_T(m), \text{ where} \tag{7}$$

$$w_R(m) = w_T(m) = [0,0,\ldots,i_m,i_{m+1},\ldots,i_{m+k-1},0,\ldots,0] \text{ and } i_m = i_{m+1} = \ldots = i_{m+k-1} = 1$$

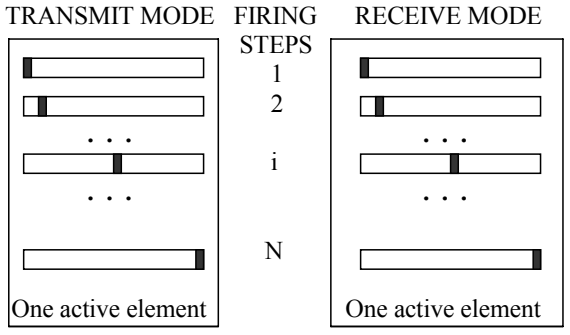


Fig.1: SAFT imaging method

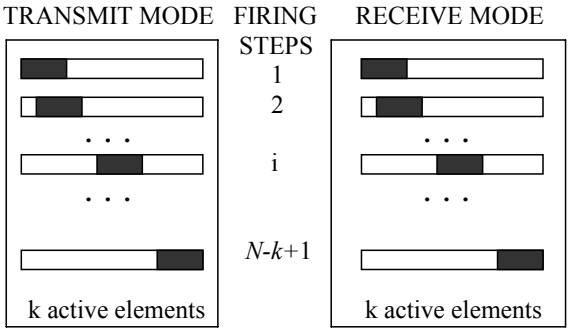


Fig.2: MSAF imaging method

In STA imaging, at each time one array element transmits a pulse and all elements receive the echo signals (Fig. 3). The advantage of this approach is that a full dynamic focusing can be applied to the transmission and

the receiving, giving the highest quality of image. The disadvantage is that a huge data memory is required and motion artifacts may occur. The effective aperture is calculated as:

$$e_N = \sum_{m=1}^{N} w_R \otimes w_T(m) \text{ where } w_R = [1,1,\ldots,1],\ w_T(m) = [0,0,\ldots,i_m,\ldots,0] \text{ and } i_m = 1 \qquad (8)$$

Synthetic Receive Aperture (SRA) method of imaging was proposed to improve lateral resolution (Fig. 4). It is known that the lateral resolution can be improved by increasing array length. In practice, it is not very expensive to build a large transmit aperture, but is very complex to form a large receive aperture. This method uses a large transmit aperture and enables an imaging system to address a large number of transducer receive elements without the same number of parallel receive channels. In the receive mode the aperture is split into two or more subapertures. In order to form each line of image data in the SRA system, the transmitters must be fired once for each receive subaperture. For a single transmit pulse (from all transmit elements), the RF sum for one receive subaperture is formed and stored in memory. Then a second identical pulse is transmitted in the same direction and the RF sum for another subaperture is formed and stored. After the RF signals have been acquired from all receive subapertures, the total RF sum is formed by coherently adding together the sums from various subapertures. For an *N*-element linear array, receive aperture is split into $N_S = N/K_R$ subapertures, and each subaperture contains K_R elements. The effective aperture is:

$$e_N = \sum_{m=0}^{N_S-1} w_R(m) \otimes w_T \text{ , where} \qquad (9)$$

$$w_T = [1,1,\ldots,1] \quad , \quad w_R(m) = [0,0,\ldots,i_n,i_{n+1},\ldots,i_{n+K_R-1},0,\ldots,0] \quad , \quad n = m * K_R + 1$$

$$i_n = i_{n+1} = \ldots = i_{n+K_R-1} = 1$$

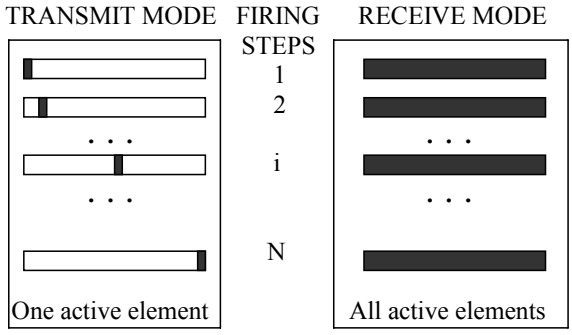


Fig.3: STA imaging method

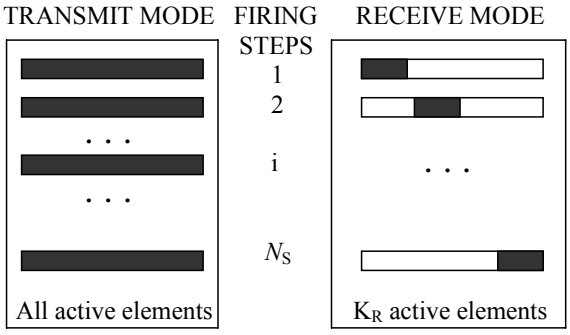


Fig.4: SRA imaging method

A sparse STA imaging method is proposed to increase system frame rate (Fig. 5). Only a small number of elements are used to transmit a pulse but all array elements receive the echo signals. For an *N*-element aperture, *M*x*N* data recordings are needed for image reconstruction., where *M*<<*N*. All data recordings must then be combined with dynamic focusing. The effective aperture is:

$$e_N = \sum_{m=1}^{M} w_T(m) \otimes w_R \qquad (10)$$

where $w_T(m) = [0,0,\ldots i_{Km},\ldots 0]$, $w_R = [1,1,\ldots,1]$ and $i_{Km} = 1$

The two-way radiation pattern of a synthetic aperture imaging system is calculated using the Fourier transform of the corresponding effective aperture function defined by the expressions (6,7,8,9 and 10).

4. Optimization of a Sparse Array

For a sparse STA imaging system with an array with *N*-elements, the two-way radiation pattern is evaluated as the Fourier Transform of the effective aperture function e_N, defined as:

$$e_N = \sum_{m=1}^{M} a_m \otimes B, \text{ and } a_m = [0,0,...,i_m,...0], \text{ where } i_m = 1 \quad (11)$$

where a_m is the transmit aperture during the *m*'th firing, *B* is the apodization function applied to the receiver elements, and $\otimes$ is the convolution operator. The speed of the image acquisition is determined by the number of transmit elements (*M*), *M*<<*N*. Since the geometrical locations of the transmit elements in a sparse array system impact the two-way radiation pattern of that system, the image quality parameters, the lateral resolution and contrast, all depend on the locations of the transmit elements within the sparse array ($i_1,i_2,\ldots,i_M$). Since the weighting applied to each receiver element also impacts the radiation pattern of the system, the image quality also depends on the type of the apodization function (*B*). Therefore, the optimization of a sparse STA imaging system can be formulated as an optimization problem of both the location of the elements of the sparse array in transmit and the weights assigned to the elements of the full array during receive [8]. Different algorithms have been proposed for optimization of the locations of the transmitted elements in a sparse array – genetic, linear programming and simulated annealing algorithms [8]. For most cases the optimization criterion is minimal sidelobe peak of the radiation pattern.

In this paper, another optimization criterion is proposed. It is the minimal width of the mainlobe (*W*) combined with a condition on the maximum sidelobe level (*SL* <Q). It is suggested here to divide the optimization process into two stages. In the first stage, the optimal positions of transmit elements (($i_1,i_2,\ldots,i_M$) are found, for a set of known apodization functions $\{B_k\}$, *k*=1,2,...*K*. Such a set of apodization functions may include several well-known window-functions (Hamming, Hann, Kaiser, Chebyshev and etc). At this stage, the optimization criterion can be written as follows:

Given M, N and $\{B\}_K$, *choose* $(i_1,i_2,\ldots,i_M)_K$ *to minimize W subject to SL*<Q (12)

where *Q* is the threshold of acceptable level of the sidelobe peak. In the second stage, the final layout of transmit elements is chosen, which is a layout that corresponds to the most appropriate apodization function *B*= ($b_1,b_2,..,b_N$). This choice is a compromise between minimal width of the mainlobe and the acceptable level of the peak of the sidelobes. Mathematically, it can be written as follows:

Given M , N, $\{B\}_K$ *and* $\{i_1,i_2,\ldots,i_M\}_K$, *choose* $(b_1,b_2,..,b_N)$ *to minimize W* subject to *SL*<Q (13)

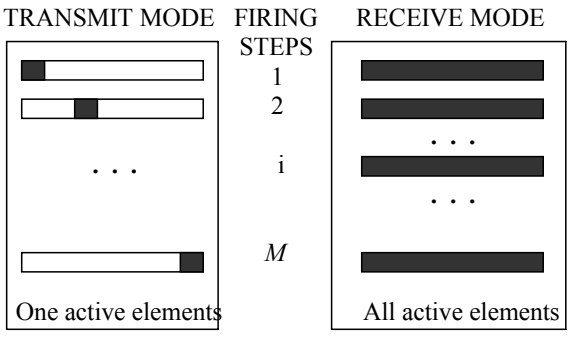


Fig.5: Sparse STA imaging method

```
begin
  Initialize (I0,T0)
  for k=1 to number_iterations
    Tk=Tk·α
    for j=1 to number_perturbations
      Ip=perturbate(Ij-1)
      ΔW=W(Ip)- W(Ij-1)
      ΔSL=SL(Ip)- Q
      if {ΔW<0 or exp(-ΔW/Tk)<rand(0,1)}
                &
         {ΔSL<0 or exp(-ΔSL/Tk)<rand(0,1)}
         Ij = Ip
      else
         Ij = Ij-1
      endif
    endfor
  endfor
end
```

Fig.6: The simulated annealing algorithm

where $\{i_1,i_2,\ldots,i_M\}_K$ are the selected positions of transmit elements, as found at the first stage of the optimization.

One way of selecting the positions $\{i_1,i_2,\ldots,i_M\}_K$ is by using a modification of the simulated annealing algorithm based on a Monte Carlo simulation. This approach was suggested initially for combinatorial optimization by Kirkpatrick et al. [9]. The simulated annealing algorithm realizes an iterative procedure that is determined by simulation of the arrays with variable transmit element positions. In order to speed up the simulation process it is assumed that two of the *M* transmit elements are always the two outer elements of the physical array; their positions are not changed and are assigned numbers 1 and *N*. The positions of the other transmit elements are shifted randomly, where a shift in position to the left or to the right has equal probability (of 0.5). Once the process is initiated, with an initial layout of transmit elements $I_0=(i^0_1,\ i^0_2,..i^0_M)$, a neighbour layout $I_1=(i^1_1,\ i^1_2,..i^1_M)$ is generated, and the algorithm accepts or rejects this layout according to a certain criterion. The acceptance is decided stochastically and may be described in terms of probability as:

$$P=\begin{cases}1 & if\ \Delta W<0\ \&\ \Delta SL<0\\ \exp(-\Delta W/T_k) & if\ \Delta W>0\ \&\ \Delta SL<0\\ \exp(-\Delta SL/T_k) & if\ \Delta W<0\ \&\ \Delta SL>0\\ \exp(-\Delta W/T_k)\times\exp(-\Delta SL/T_k) & if\ \Delta W>0\ \&\ \Delta SL>0\end{cases} \tag{14}$$

where *P* is the probability of acceptance, ΔW is the difference of width of the mainlobe, ΔSL is the difference of the height of the peak of the sidelobe between the current configuration of transmit elements and the best one obtained at preceding steps. T_k is the current value of 'temperature', where the current 'temperature' is evaluated as $T_k=0.95T_{k-1}$, and the algorithm proceeds until the number of iterations reaches the final value. A pseudo-code of the proposed simulated annealing algorithm is given in Fig.6.

5. Computations and Comparison Analysis

The effective aperture function and the corresponding two-way radiation pattern of several ultrasound imaging systems were calculated using Eqns (1-14), in order to compare the quality of images produced by the questioned systems.

Synthetic aperture. Three more perspective types of synthetic aperture imaging systems are investigated.

The investigated MSAF imaging system employs a linear array with 64 elements and active sub-apertures with 4, 8 (Fig. 7), 16 (Fig. 8) or 32 elements, respectively. In the study no apodization is used. The best results for the lateral resolution and *SL* are obtained when the active transmit sub-aperture consists of only 4 elements (Table 1). The main disadvantage of the small number of active transmit elements is that the transmitted power is less, hence the SNR is low.

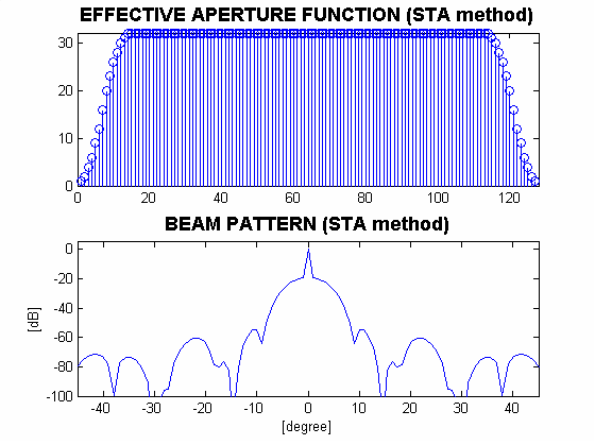


Fig. 7: MSAF method with 8 active elements

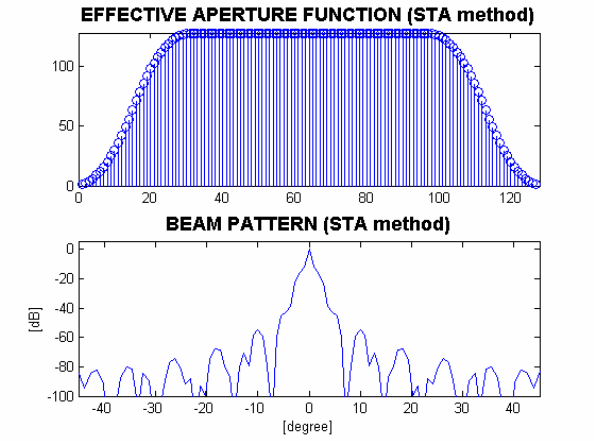


Fig. 8: MSAF method with 16 active elements

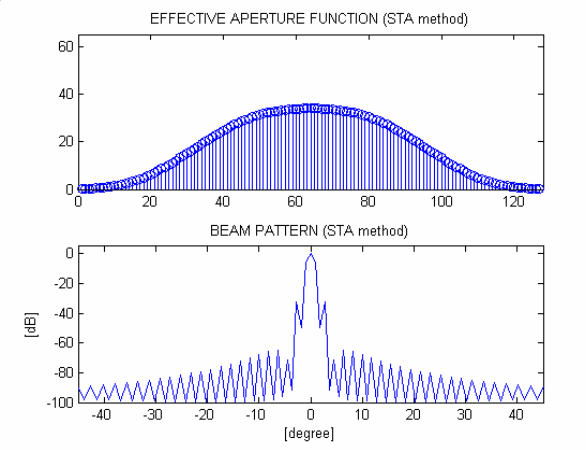


Fig.9 Conventional STA method
(64-elements, Hamming apodization)

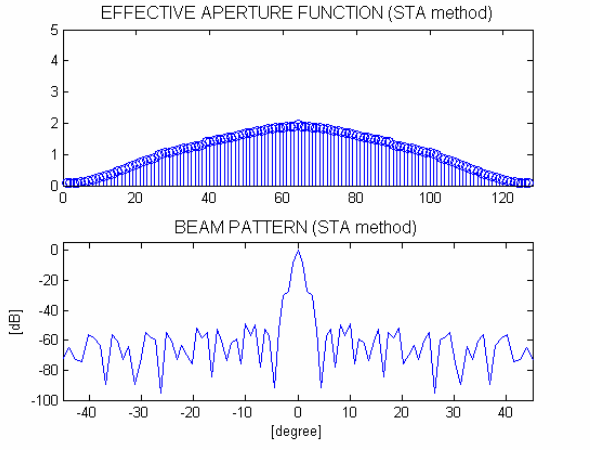


Fig. 10: Sparse STA method
(4 transmit elements, 64 receive elements,
Hamming apodization)

Number of transmit elements	4	8	16	32
Δθ °	0.3896	0.5664	0.9188	1.7685
SL, dB	-59.6213	-54.3839	-54.574	-35.2764

Table 1: Mainlobe and sidelobe peak level of a MSAF imaging system

In a conventional STA imaging method, the transducer is of 64 elements. In the receive mode all elements are active. In the transmit mode, the linear array is split into 64 sub-apertures, each of them has only one active element. The effective aperture function and the corresponding two-way radiation pattern of such a STA imaging system is shown in Fig.9. The –6dB beamwidth obtained for the STA imaging system is 1.82º. However, the sidelobe peak level of the STA system is only –32 dB.

Sparse array optimization. Computer simulations were performed in order to optimize the design and performance of a sparse array probe, to be used for synthetic transmit aperture imaging. The example given here is of a 64 elements sparse array, where 64 active elements are used in receive and only 4 elements are used in transmit. The properties of the system are optimized using the two-stage algorithm described in section 4. First, the optimal positions of transmit elements are found for three apodization functions – Boxcar (i.e. no apodization of the receiver elements), Hamming and the Blackman-Harris.

Optimized positions of transmit elements	Receiver Apodization	Mainlobe width (Δθ °)			*SL* (dB)
		-6 dB	-20 dB	-40 dB	
1, 2, 63, 64	-	0.33	1.11	3.2	-33
1, 26, 39, 64	Hamming	1.34	2.92	6.2	-50
1, 21, 44, 64	Blackman-Harris	1.32	4.33	11.16	-100

Table 2: Numerical results obtained after employing the two stages of the optimization.

For each apodization function, the positions of transmit elements are shifted until optimal performance is obtained, as described earlier, using the simulated annealing algorithm presented in Fig.6. In order to obtain a radiation pattern with a sharper mainlobe, the optimization criterion was formulated as the minimal width of

the mainlobe at –20dB (instead of at –6 dB) below the maximum where the condition that the maximal level of the sidelobe peak is below –50 dB. The positions of transmit elements that were found to optimize the performance of the system, studied for a physical array with $\lambda/2$ element spacing, together with the achieved widths of the mainlobe (at -6 dB, -20 dB and –40 dB) and the levels of the peaks of the sidelobe, are all presented in Table 2. Both optimized functions, the effective aperture function and the corresponding two-way radiation pattern, are plotted for Hamming apodization function (Fig. 10). It may be seen that the apodization reduces the levels of the peaks of the sidelobes from –33 dB to –100 dB, but at the cost of widening the mainlobe of the radiation pattern. Since the dynamic range of a computer monitor is limited to about 50 dB, the sparse array is chosen with the Hamming apodization and the locations of the transmit elements are set to be at positions 1, 26, 39 and 64. Comparison analysis of numerical results (Table 2, Fig.9) shows that a sparse STA imaging system improves significantly lateral resolution of images because $\Delta\theta=1.82^{\circ}$ - for a conventional STA imaging system and $\Delta\theta=1.34^{\circ}$ - for a sparse STA imaging system.

6. Conclusions

It is shown that the effective aperture approach can be successfully used as a tool for analysis and parameter optimization of the synthetic aperture imaging systems. The effective aperture function and the corresponding two-way radiation function provide information about two of the most important parameters of images produced by an ultrasound system - lateral resolution and contrast. Therefore, in the design, optimization of the effective aperture function leads to optimal choice of such parameters of a STA imaging system that influence on lateral resolution and contrast of images produced by this imaging system.

The numerical results show that each system has its own advantages and disadvantages. The choice of imaging system should depend on the task, which it will be used for. It is shown that Hamming apodization gives the best compromise between the contrast of images and the lateral resolution.

A MSAF system has better lateral resolution and SL level with less active elements, but in that case the SNR is lowered.

The sparse synthetic transmit aperture imaging systems can be proposed as an alternative and superior approach to the conventional STA systems. Yet, the sparse STA imaging systems suffer from some deficiencies. With proper design, these deficiencies can be overcome and the sparse STA imaging system can perform extremely well for specific applications. To do so, an effective aperture approach is used for optimization of the sparse STA imaging system. A two-stage algorithm is proposed for optimizing both the locations of transmit elements within the ultrasound probe and the weights of the receive element. The first stage of the optimization procedure employs a simulated annealing algorithm that optimizes the locations of the transmit elements for a set of apodization functions. At the second stage, an appropriate apodization function is selected.

Acknowledgments

This work was supported by the Centre of Excellence BIS21++ and the Bulgarian National Science Fund – grants I-1202/02, I-1205/02 and MI-1506/05.

Bibliography

B. Angelsen, Ultrasound imaging: Waves, signals, and signal processing, Emantec, Norway, 2000.

S. Holm and H. Yao, Method and apparatus for synthetic transmit aperture imaging, US patent No 5.951.479, Sep. 14, 1999.

Ylitalo, On the signal-to-noise ratio of a synthetic aperture ultrasound imaging method, Europ. J. Ultras. 3, (1996), 277 - 281.

M.Karaman, H. Bilge, and M. O'Donnell, Adaptive multi-element synthetic aperture imaging with motion and phase aberration correction, IEEE Trans. Ultrason. Ferroelec. Freq. Contr., vol. 45, 4, (1998), 1077-1087.

G. Trahey, and L. Nock, Multi-element synthetic transmit aperture imaging using temporal coding, IEEE Trans. Med. Imag., vol. 22, 4, (2003), 552-563.

V. Behar, and D. Adam, Optimization of sparse synthetic transmit aperture imaging with coded excitation and frequency division, Ultrasonics, (2005) , (submitted to be printed)

G. R Lockwood and F. S. Foster, Design of Sparse Array Imaging Systems

S. Holm, A. Austeng, K. Iranpour, J. Hopperstad, Sparse sampling in array processing, Chapter 19 in "Sampling theory and practice", (F. Marvasti Ed.), Plenum, N.Y., (2001)

S. Kirkpatrik, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science, vol. 220, 4598, (1988), 671-680.

Authors' Information

Milen Nikolov – Institute for Parallel Processing, Bulgarian Academy of Science, Acad. G. Bonchev Str., 25-A, Sofia 1113, Bulgaria, e-mail: milenik@bas.bg

Vera Behar – Institute for Parallel Processing, Bulgarian Academy of Science, Acad. G. Bonchev Str., 25-A, Sofia 1113, Bulgaria, e-mail: behar@bas.bg

A MATHEMATICAL APPARATUS FOR ONTOLOGY SIMULATION. SPECIALIZED EXTENSIONS OF THE EXTENDABLE LANGUAGE OF APPLIED LOGIC[1]

Alexander Kleshchev, Irene Artemjeva

Abstract: *A mathematical apparatus for domain ontology simulation is described in the series of articles. This article is the second one of the series. It describes a few specialized extensions of the extendable languages of applied logic that was described in the first article of the series. A few examples of some ideas related to domain ontologies and formalization of these ideas using the language are presented.*

Keywords*: Extendable language of applied logic, ontology language specification, specialized extensions of the extendable language of applied logic.*

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

Introduction

The definition of the extendable language of applied logic was given in [Kleshchev et al, 2005]. This definition consists of the kernel of the language and of its standard extension only. When the semantic basis is extended for particular applications the following two classes of elements are possible. The elements of the first class can be impossible or undesirable to be defined by means of the kernel of the language and by extensions built. On the contrary, the elements of the second class can be naturally defined by means of the kernel and extensions built. The elements of the first class are described in specialized extensions in the same form that is used in the description of the kernel of the language and of its standard extension. A specialized extension of the language defines elements of the semantic basis that are necessary for a comparatively narrow class of applications. Because the same specialized extensions can be used in different applications such extensions have names. Every particular language of applied logic contains the kernel and usually the standard extension and possibly some specialized extensions. By this means, every particular language of applied logic is characterized by a set of extension names rather than a signature. A signature is introduced by a particular logical theory represented by

[1] This paper was made according to the program of fundamental scientific research of the Presidium of the Russian Academy of Sciences «Mathematical simulation and intellectual systems», the project "Theoretical foundation of the intellectual systems based on ontologies for intellectual support of scientific researches".

such a language. Therewith, propositions of the theory can associate values (interpretation) or sorts with names (elements of the signature) or can restrict possible functions of interpretations for these names according to the interpretation of other names. In turn, every theory has a name. The parameters of the name are the names of the extensions of the language that are used for describing the theory. Other theories represented by their names also can be elements of the theory.

This article describes a few specialized extensions of the languages and a few examples of some ideas related to domain ontologies and formalization of these ideas using the language.

1. Specialized Extension "Intervals" of the Language of Applied Logic

Every specialized extension of the language has a name. In the representation of a logical theory it must be indicated which extensions of the language are used in this representation. In this paragraph the specialized extension *Intervals* is defined.

The terms of the extension are:

1. [ ]R, and also $J_{\alpha\theta}$([ ]R) is the set of all possible intervals of real numbers; $J_{\alpha\theta}$([ ]R) does not depend on an interpretation function α and on an admissible substitution θ;

2. R[t_1, t_2], where t_1 and t_2 are terms; $J_{\alpha\theta}$(R[t_1, t_2]) is the set of all the real numbers which are not less than $J_{\alpha\theta}(t_1)$ and are not greater than $J_{\alpha\theta}(t_2)$; the value of the term exists if both $J_{\alpha\theta}(t_1)$ and $J_{\alpha\theta}(t_2)$ are numbers and $J_{\alpha\theta}(t_1) \leq J_{\alpha\theta}(t_2)$;

3. R(t_1, t_2], where t_1 and t_2 are terms; $J_{\alpha\theta}$(R(t_1, t_2]) is the set of all the real numbers which are greater than $J_{\alpha\theta}(t_1)$ and are not greater than $J_{\alpha\theta}(t_2)$; the value of the term exists if both $J_{\alpha\theta}(t_1)$ and $J_{\alpha\theta}(t_2)$ are numbers and $J_{\alpha\theta}(t_1) < J_{\alpha\theta}(t_2)$;

4. R[t_1, t_2), where t_1 and t_2 are terms; $J_{\alpha\theta}$(R[t_1, t_2)) is the set of all the real numbers which are not less than $J_{\alpha\theta}(t_1)$ and are less than $J_{\alpha\theta}(t_2)$; the value of the term exists if both $J_{\alpha\theta}(t_1)$ and $J_{\alpha\theta}(t_2)$ are numbers and $J_{\alpha\theta}(t_1) < J_{\alpha\theta}(t_2)$;

5. R(t_1, t_2), where t_1 and t_2 are terms; $J_{\alpha\theta}$(R(t_1, t_2))is the set of all the real numbers which are greater than $J_{\alpha\theta}(t_1)$ and are less than $J_{\alpha\theta}(t_2)$; the value of the term exists if both $J_{\alpha\theta}(t_1)$ and $J_{\alpha\theta}(t_2)$ are numbers and $J_{\alpha\theta}(t_1) < J_{\alpha\theta}(t_2)$;

6. R[t, ∞), where t is a term; $J_{\alpha\theta}$(R[t, ∞)) is the set of all the real numbers which are not less than $J_{\alpha\theta}$(t); the value of the term exists if $J_{\alpha\theta}$(t) is a number;

7. R(t, ∞), where t is a term; $J_{\alpha\theta}$(R(t, ∞)) is the set of all the real numbers which are greater than $J_{\alpha\theta}$(t); the value of the term exists if $J_{\alpha\theta}$(t) is a number;

8. R(-∞, t], where t is a term; $J_{\alpha\theta}$(R(-∞, t]) is the set of all the real numbers which are not greater than $J_{\alpha\theta}$(t); the value of the term exists if $J_{\alpha\theta}$(t) is a number;

9. R(-∞, t), where t is a term; $J_{\alpha\theta}$(R(-∞, t)) is the set of all the real numbers which are less than $J_{\alpha\theta}$(t); the value of the term exists if $J_{\alpha\theta}$(t) is a number;

10. I and also $J_{\alpha\theta}$(I) is the set of all the integers; $J_{\alpha\theta}$(I) does not depend on α and θ;

11. [ ]I and also $J_{\alpha\theta}$([ ]I) is the set of all possible intervals of integers; $J_{\alpha\theta}$([ ]I) does not depend on α and θ;

12. I[t_1, t_2],where t_1 and t_2 are terms; $J_{\alpha\theta}$(I[t_1, t_2]) is the set of all the integers which are not less than $J_{\alpha\theta}(t_1)$ and are not greater than $J_{\alpha\theta}(t_2)$; the value of the term exists if $J_{\alpha\theta}(t_1)$ and $J_{\alpha\theta}(t_2)$ are numbers and $J_{\alpha\theta}(t_1) \leq J_{\alpha\theta}(t_2)$;

13. I[t, ∞), where t is a term; $J_{\alpha\theta}$(I[t, ∞)) is the set of all the integers which are not less than $J_{\alpha\theta}$(t); the value of the term exists if $J_{\alpha\theta}$(t) is a number;

14. I(-∞, t], where t is a term; $J_{\alpha\theta}$(I(-∞, t]) is the set of all the integers which are not greater than $J_{\alpha\theta}$(t); the value of the term exists if $J_{\alpha\theta}$(t) is a number;

15. inf(t), where t is a term; $J_{\alpha\theta}$(inf(t)) is the minimal element of the set $J_{\alpha\theta}$(t); the value of the term exists if $J_{\alpha\theta}$(t) is a set of numbers that has the minimal element;

16. sup(t), where t is a term; $J_{\alpha\theta}$(sup(t) is the maximal element of the set $J_{\alpha\theta}$(t); the value of the term exists if $J_{\alpha\theta}$(t) is a set of numbers that has the maximal element.

The extension defines no new types of formulas.

3. Specialized Extension "Mathematical Quantifiers" of the Language of Applied Logic.

The terms of the extension are:

1. a quantifier construction (Σ (v_1: t_1)…(v_m: t_m) t) (quantifier of summation); $J_{\alpha\theta}$((Σ (v_1: t_1)…(v_m: t_m) t)) is equal to the sum of the values $J_{\alpha\theta}$(t), where θ belongs to the set of admissible substitutions for (v_1: t_1)…(v_m: t_m); the value of the term exists if $J_{\alpha\theta}$(t) is a number for every admissible substitution θ for (v_1: t_1)…(v_m: t_m);

2. a quantifier construction (Π (v_1: t_1)…(v_m: t_m) t) (quantifier of multiplication); $J_{\alpha\theta}$((Π ((v_1: t_1)…(v_m: t_m) t)) is equal to the product of the values $J_{\alpha\theta}$(t), where θ belongs to the set of admissible substitutions for (v_1: t_1)…(v_m: t_m); the value of the term exists if $J_{\alpha\theta}$(t) is a number for every admissible substitution θ for (v_1: t_1)…(v_m: t_m);

3. a quantifier construction ($\cup$ (v_1: t_1)…(v_m: t_m) t) (quantifier of union); $J_{\alpha\theta}$(($\cup$ (v_1: t_1)…(v_m: t_m) t)) is equal to the union of the values $J_{\alpha\theta}$(t), where θ belongs to the set of admissible substitutions for (v_1: t_1)…(v_m: t_m); the value of the term exists if $J_{\alpha\theta}$(t) is a set for every admissible substitution θ for (v_1: t_1)…(v_m: t_m);

4. a quantifier construction ($\cap$ (v_1: t_1)…(v_m: t_m) t) (quantifier of intersection); $J_{\alpha\theta}$(($\cap$ (v_1: t_1)…(v_m: t_m) t)) is equal to the intersection of the values $J_{\alpha\theta}$(t), where θ belongs to the set of admissible substitutions for (v_1: t_1)…(v_m: t_m); the value of the term exists if $J_{\alpha\theta}$(t) is a set for every admissible substitution θ for (v_1: t_1)…(v_m: t_m).

The formulas of the extension are:

1. a quantifier construction (& (v_1: t_1)…(v_m: t_m) f) (quantifier of conjunction); $J_{\alpha\theta}$((& (v_1: t_1)…(v_m: t_m) f)) is true if and only if all the values $J_{\alpha\theta}$(f) are true when θ belongs to the set of admissible substitutions for (v_1: t_1)…(v_m: t_m); the formula has a value if $J_{\alpha\theta}$(f) has a value for every admissible substitution θ for (v_1: t_1)…(v_m: t_m);

2. a quantifier construction ($\vee$ (v_1: t_1)…(v_m: t_m) f) (quantifier of disjunction); $J_{\alpha\theta}$(($\vee$ (v_1: t_1)…(v_m: t_m) f)) is true if and only if at least one of the values $J_{\alpha\theta}$(f) is true when θ belongs to the set of admissible substitutions for (v_1: t_1)…(v_m: t_m); the formula has a value if $J_{\alpha\theta}$(f) has a value for every admissible substitution θ for (v_1: t_1)…(v_m: t_m).

4. Examples of Applied Logical Theories and Their Models

Here a few examples of some ideas related to domain ontologies and formalization of these ideas will be presented.

Example 1. An applied logical theory "Definition of partitions (ST, Intervals, Mathematical quantifiers)". This applied logical theory contains only value descriptions for names.

(1.1.1) partitions ≡ ($\cup$ (n: I[1, ∞)) {(v: R $\Uparrow$ (n + 1)) (& (i : I[1, n]) π(i, v) < π(i + 1, v))}

"Partitions" means the set of all possible partitions of the set of real numbers into intervals; every partition is a finite strictly increasing sequence of numbers.

(1.1.2) element ≡ (λ (partition: partitions) (i: I[0, length(partition) - 1]) π(i + 1, partition))

"Element" is a function; its arguments are a partition v and an integer i in the range from 0 to the number of elements in the partition v; its result is the i-th element of the partition v.

(1.1.3) interval ≡ (λ (partition : partitions) (i: I[1, length(partition)-1]) R[element(partition, i-1), element(partition, i)])

"Interval" is a function; its arguments are a partition v and an integer i in the range from 0 to the number of elements in the partition v; its result is the interval consisting of all the real numbers between the (i - 1)-th and the i-th elements of the partition v.

It is obvious that this applied logical theory has no models since it contains no ambiguously interpreted names.

Example 2. An applied logical theory “T1(ST, Intervals, Mathematical quantifiers)”, represents a model for a simplified ontology of medical diagnostics: T1(ST, Intervals, Mathematical quantifiers) = <{Definition of partitions}, SS>, where SS is the following set of propositions.

The value descriptions for names

(2.1.1) sets of values ≡ ({ } N) ∪ ([ ]I) ∪ ([ ]R)

“Sets of values” means the set of possible value ranges for all signs; these ranges can be sets of names (ranges of qualitative values), integer-valued and real-valued intervals (ranges of quantitative values).

The sort descriptions for names.

(2.2.1) sort signs: { }N

“Signs” means a finite set of medical sign names.

(2.2.2) sort diseases: { }N

“Diseases” means a finite set of disease names.

(2.2.3) sort possible values: signs → sets of values

“Possible values” means a function that takes a sign and returns its possible value range.

(2.2.4) sort normal values: signs → sets of values

“Normal values” means a function that takes a sign and returns its normal value range.

(2.2.5) sort clinical picture: diseases → { } signs

“Clinical picture” is a function that takes a disease and returns a subset of the set of signs, which is the clinical picture of the disease.

(2.2.6) sort number of dynamics periods: {(disease: diseases) (sign: clinical picture(disease))} → I[1, ∞)

“Number of dynamics periods” is a function that takes a disease and a sign from the clinical picture of the disease and returns the number of dynamics periods of the sign for the disease.

(2.2.7) sort values for a dynamics period: {(disease: diseases) (sign: clinical picture(disease)) (index of dynamics period: I[1, number of dynamics periods(disease, sign)])} → sets of values

“Values for a dynamics period” means a function that takes a disease, a sign from the clinical picture of the disease and an index of a dynamics period of the sign for the disease and returns a set of values of the sign, which are possible during the dynamics period.

(2.2.8) sort upper bound: {(disease: diseases) (sign: clinical picture(disease)) (index of dynamics period: I[1, number of dynamics periods(disease, sign)])} → I[0, ∞)

“Upper bound” is a function that takes a disease, a sign from the clinical picture of the disease and an index of a dynamics period of the sign for the disease and returns an upper bound of the duration of the dynamics period.

(2.2.9) sort lower bound: {(disease: diseases) (sign: clinical picture(disease)) (index of dynamics period: I[1, number of dynamics periods(disease, sign)])} → I[0, ∞)

“Lower bound” is a function that takes a disease, a sign from the clinical picture of the disease and an index of a dynamics period of the sign for the disease and returns a lower bound of the duration of the dynamics period.

(2.2.10) sort diagnosis: diseases

“Diagnosis” is the disease which the patient is ill with; in this model diagnosis can be either a disease or healthy.

(2.2.11) sort partition for a sign: clinical picture(diagnosis) → partitions

“Partition for a sign” is a function that takes a sign from the clinical picture of the disease, which the patient is ill with and returns a partition of the patient's time axis.

(2.2.12) sort moments of examination: signs → { }I[0, ∞)

Moments of examination means a function that takes a sign and returns a set of time moments at which the sign of the patient was examined; the time is measured by an integer amount of hours from the beginning of the patient's examination.

(2.2.13) (sign: signs) sort sign: moments of examination(sign) → possible values(sign)

Every term belonging to set “signs” means a function (process) that takes a moment of examination of the sign and returns the value of the patient's sign at the moment; any value of this kind is a possible value of the sign.

The restrictions on the interpretation of names.

(2.3.1) (sign: signs) (normal values(sign) $\neq \varnothing$) & (normal values(sign) $\subset$ possible values(sign))

For any sign its set of normal values is a nonempty proper subset of its set of possible values.

(2.3.2) clinical picture(healthy) = $\varnothing$

Clinical picture of healthy contains no signs.

(2.3.3) (disease: diseases) (sign: clinical picture(disease)) (index of dynamics period: I[1, number of dynamics periods(disease, sign)]) (values for a dynamics period(disease, sign, index of dynamics period) $\neq \varnothing$) & (values for a dynamics period(disease, sign, index of dynamics period) $\subseteq$ possible values(sign)) & (upper bound(disease, sign, index of dynamics period) > lower bound(disease, sign, index of dynamics period))

For any disease, for any sign from the clinical picture of the disease and for any dynamics period of the sign, the set of values of the sign possible in the dynamics period is a nonempty subset of the set of possible values of the sign; upper bound of the dynamics period is greater than its lower bound.

(2.3.4) (disease: diseases) (sign: clinical picture(disease)) ($\vee$ (index of dynamics period: I[1, number of dynamics periods(disease, sign)]) values for a dynamics period (disease, sign, index of dynamics period) $\cap$ (possible values(sign) \ normal values(sign)) $\neq \varnothing$)

For any disease and for any sign from the clinical picture of the disease, the set of values of the sign possible at least in one dynamics period contains values, which are not normal for the sign.

(2.3.5) (sign: signs \ clinical picture(diagnosis)) (moment of examination: moments of examination(sign)) sign(moment of examination) $\in$ normal values(sign)

For any sign not belonging to the clinical picture of the disease the patient is ill with, the value of the sign can be only normal at any time moment.

(2.3.6) (sign: clinical picture(diagnosis)) length(partition for a sign(sign)) = number of dynamics periods(diagnosis, sign) + 1

For any sign from the clinical picture of the disease the patient is ill with, the number of intervals in the patient's partition for the sign is equal to the number of dynamics periods for the sign and disease.

(2.3.7) (sign: clinical picture(diagnosis)) (index of dynamics period: I[1, number of dynamics periods(diagnosis, sign)]) (moment of examination: moments of examination(sign) $\cap$ interval(partition for a sign(sign), index of dynamics period)) sign (moment of examination) $\in$ values for a dynamisc period(diagnosis, sign, index of dynamics period)

For any sign from the clinical picture of the disease the patient is ill with, for any dynamics period of the sign and for any moment of examination belonging to the dynamics period, the value of the sign examined at the moment is a possible value for the dynamics period.

(2.3.8) (sign: clinical picture(diagnosis)) (index of dynamics period: I[1, number of dynamics periods(diagnosis, sign)]) sup(interval(partition for a sign(sign), index of dynamics period)) – inf(interval(partition for a sign(sign), index of dynamics period)) $\in$ R[lower bound(diagnosis, sign, index of dynamics period), upper bound(diagnosis, sign, index of dynamics period)]

For any sign from the clinical picture of the disease the patient is ill with and for any dynamics period of the sign, the duration of the dynamics period is greater than the lower bound and less than the upper bound of the dynamics period.

Example 3. A model of the applied logical theory of example 2 represented by a set of value descriptions for names.

(3.1.1) signs $\equiv$ {strain of abdomen muscles, blood pressure, daily diuresis}

Only three signs are considered: strain of abdomen muscles, blood pressure and daily diuresis.

(3.1.2) diseases ≡ {healthy, pancreatitis}

Only two diseases (states) are considered: healthy and pancreatitis.

(3.1.3) possible values ≡ (λ (sign: {strain of abdomen muscles, blood pressure, daily diuresis}) /(sign = strain of abdomen muscles ⇒ {presence, absence}), (sign ∈ {blood pressure, daily diuresis} ⇒ {normal, high, low})/)

The possible values of strain of abdomen muscles are presence and absence; those of blood pressure and daily diuresis are normal, high and low.

(3.1.4) normal values ≡ (λ (sign: {strain of abdomen muscles, blood pressure, daily diuresis}) /(sign = strain of abdomen muscles ⇒ {absence}), (sign ∈ {blood pressure, daily diuresis} ⇒ {normal})/)

The normal value of strain of abdomen muscles is absence; that of blood pressure and daily diuresis is normal.

(3.1.5) clinical picture ≡ (λ (disease: {healthy, pancreatitis}) /(disease = healthy ⇒ ∅) (disease = pancreatitis ⇒ { strain of abdomen muscles, blood pressure, daily diuresis})/)

The clinical picture of healthy is empty; the one of pancreatitis consists of strain of abdomen muscles, blood pressure and daily diuresis.

(3.1.6) number of dynamics periods ≡ (λ (v: {<pancreatitis, strain of abdomen muscles>, <pancreatitis, blood pressure>, <pancreatitis, daily diuresis>}) /(π(1,v) = pancreatitis & π(2,v) ∈ {strain of abdomen muscles, blood pressure, daily diuresis} ⇒ 2)/)

For pancreatitis the number of dynamics periods of strain of abdomen muscles, blood pressure and daily diuresis is equal to 2.

(3.1.7) values for a dynamics period ≡ (λ (v:{<pancreatitis, strain of abdomen muscles,1>, <pancreatitis, strain of abdomen muscles, 2>, <pancreatitis, blood pressure, 1>, < pancreatitis, blood pressure, 2>, <pancreatitis, daily diuresis, 1>, <pancreatitis, daily diuresis, 2>}) /(v=<pancreatitis, strain of abdomen muscles,1> ⇒ {absence}), (v = <pancreatitis, strain of abdomen muscles, 2> ⇒ {presence}), (v = <pancreatitis, blood pressure,1> ⇒ {normal}), (v = <pancreatitis, blood pressure, 2> ⇒ {high}), (v = <pancreatitis, daily diuresis, 1> ⇒ {low}), (v = <pancreatitis, daily diuresis,2> ⇒ {normal})/)

For pancreatitis the value of strain of abdomen muscles in the first dynamics period can be only absence;in the second dynamics period the one can be only presence; the value of blood pressure in the first dynamics period can be only normal; in the second dynamics period the one can be only high; the value of daily diuresis in the first dynamics period can be only low; in the second dynamics period the one can be only normal.

(3.1.8) upper bound≡ (λ (v: {<pancreatitis, strain of abdomen muscles, 1>, <pancreatitis, strain of abdomen muscles, 2>, < pancreatitis, blood pressure, 1>, < pancreatitis, blood pressure, 2>, <pancreatitis, daily diuresis, 1>, <pancreatitis, daily diuresis, 2>}) /(v = <pancreatitis, strain of abdomen muscles, 1> ⇒ 48), (v = <pancreatitis, strain of abdomen muscles, 2> ⇒ 144), (v = <pancreatitis, blood pressure, 1> ⇒ 24), (v = <pancreatitis, blood pressure, 2> ⇒ 144), (v = <pancreatitis, daily diuresis, 1> ⇒ 72), (v = <pancreatitis, daily diuresis, 2> ⇒ 144)/)

For pancreatitis the upper bound of the first dynamics period of strain of abdomen muscles is equal to 48; the one of the second dynamics period is equal to 144; the upper bound of the first dynamics period of blood pressure is equal to 24; the one of the second dynamics period is equal to 144; the upper bound of the first dynamics period of daily diuresis is equal to 72; the one of the second dynamics period is equal to 144.

(3.1.9) lower bound ≡ (λ (v: {<pancreatitis, strain of abdomen muscles, 1>, <pancreatitis, strain of abdomen muscles, 2>, <pancreatitis, blood pressure, 1>, <pancreatitis, blood pressure, 2>, <pancreatitis, daily diuresis, 1>, <pancreatitis, daily diuresis, 2>}) /(v = <pancreatitis, strain of abdomen muscles, 1> ⇒ 24), (v = <pancreatitis, strain of abdomen muscles, 2> ⇒ 1), (v = <pancreatitis, blood pressure, 1> ⇒ 1), (v = <pancreatitis, blood pressure, 2> ⇒ 1), (v = < pancreatitis, daily diuresis, 1> ⇒ 48), (v = <pancreatitis, daily diuresis, 2> ⇒ 1)/)

For pancreatitis the lower bound of the first dynamics period of strain of abdomen muscles is equal to 24; the one of the second dynamics period is equal to 1; the lower bound of the first dynamics period of blood pressure is equal to 1; the one of the second dynamics period is equal to 1; the lower bound of the first dynamics period of daily diuresis is equal to 48; the one of the second dynamics period is equal to 1.

(3.1.10) diagnosis ≡ pancreatitis

The diagnosis of the patient is pancreatitis.

(3.1.11) partition for a sign≡ (λ (sign: {strain of abdomen muscles, blood pressure, daily diuresis}) /(sign = strain of abdomen muscles ⇒ <0, 40, 70>), (sign = blood pressure ⇒ <0, 20, 70>), (sign = daily diuresis ⇒ <0, 50, 70>)/)

The first dynamics period of strain of abdomen muscles is completed in 40 hours and the second one is completed in 70 hours after the beginning of the patient's examination; the first dynamics period of blood pressure is completed in 20 hours and the second one is completed in 70 hours after the beginning of the patient's examination; the first dynamics period of daily diuresis is completed in 50 hours and the second one is completed in 70 hours after the beginning of the patient's examination.

(3.1.12) moments of examination ≡ (λ (sign: {strain of abdomen muscles, blood pressure, daily diuresis}) /(sign = strain of abdomen muscles ⇒ {12,36,60}), (sign = blood pressure ⇒ {12,60}), (sign = daily diuresis ⇒ {36,60})/)

Strain of abdomen muscles is examined in 12, 36 and 60 hours after the beginning of the patient's examination; blood pressure is examined in 12 and 60 hours after the beginning of the patient's examination; daily diuresis is examined in 36 and 60 hours after the beginning of the patient's examination.

(3.1.13) strain of abdomen muscles ≡ (λ (moment of examination: {12,36,60}) /(moment of examination ∈ {12, 36} ⇒ absence), (moment of examination = 60 ⇒ presence)/)

In 12 and 36 hours after the beginning of the patient's examination the value of strain of abdomen muscles is absence; in 60 hours after the beginning of the patient's examination its value is presence.

(3.1.14) blood pressure ≡ (λ (moment of examination: {12, 60}) /(moment of examination = 12 ⇒ normal), (moment of examination = 60 ⇒ high)/)

In 12 hours after the beginning of the patient's examination the value of blood pressure is normal; in 60 hours after the beginning of the patient's examination its value is high.

(3.1.15) daily diuresis ≡ (λ (moment of examination: {36,60}) /(moment of examination = 36 ⇒ low), (moment of examination = 60 ⇒ normal)/)

In 36 hours after the beginning of the patient's examination the value of daily diuresis is low; in 60 hours after the beginning of the patient's examination its value is normal.

Conclusions

In this article a few specialized extensions for the language of applied logic have been described. Every specific language is characterized by a set (perhaps empty) consisting of the standard extension and specialized extensions. Also a few examples of some ideas related to domain ontologies and formalization of these ideas using the language have been presented.

References

[Kleshchev et al, 2005] Kleshchev A.S. and Artemjeva I.L. A mathematical apparatus for ontology simulation. An extendable language of applied logic. // In International Journal Information Theories & Applications. 2005.

Authors' Information

Alexander S. Kleshchev – kleschev@iacp.dvo.ru

Irene L. Artemjeva – artemeva@iacp.dvo.ru

Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences

5 Radio Street, Vladivostok, Russia

SERVICES FOR SATELLITE DATA PROCESSING

Andriy Shelestov, Oleksiy Kravchenko, Michael Korbakov

***Abstract**: Data processing services for Meteosat geostationary satellite are presented. Implemented services correspond to the different levels of remote-sensing data processing, including noise reduction at preprocessing level, cloud mask extraction at low-level and fractal dimension estimation at high-level. Cloud mask obtained as a result of Markovian segmentation of infrared data. To overcome high computation complexity of Markovian segmentation parallel algorithm is developed. Fractal dimension of Meteosat data estimated using fractional Brownian motion models.*

***Keywords**: cloud mask, fractals, Meteosat, Markov Random Fields, fractional Brownian motion, parallel programming, MPI.*

***ACM Classification Keywords**: I.4.6 Image Processing and Computer Vision: Segmentation – Pixel classification, G.1.2 [Numerical Analysis]: Approximation – Wavelets and fractals, D.1.3 Programming Techniques: Concurrent Programming – Parallel programming, I.4.7 Image Processing and Computer Vision: Feature Measurement – Texture*

Introduction

Among the variety of artificial Earth satellites geostationary satellites stand out due to their unique capability to observe Earth in a high frequency manner. The achievement of similar temporal characteristics using low orbit satellite platform could be possible only with a fair amount of satellites. However geostationary satellites suffer from two main drawbacks – low spatial resolution and relatively small amount of spectral bands. By these circumstances it is common to use geostationary satellites in the investigation of global Earth processes especially in the field of meteorology.

In this paper three services for geostationary satellite data processing are described. These services are designed to process data of Meteosat satellite that is operated by EUMETSAT international organization and provides information for solving practical meteorological problems. This satellite's onboard equipment makes one image of earth disk in 30 minutes in three spectral bands – visible (VIS), infrared (IR), water vapour (WV). Developed services include preprocessing service of noise detection and reduction, service for cloud mask extraction and high-level service for fractal features estimation.

Although noise reduction as preprocessing step is obviously important for further processing one of the most useful satellite data products is cloud mask. It can be used in a standalone way in applications such as air flights and satellite photography planning. Also it can be used as an input data for various satellite data processing algorithms like Normalized Difference Vegetation Index (NDVI), Sea Surface Temperature (SST) and operational wind vectors maps extraction, or even more complex applications such as numerical weather models.

A common approach for cloud mask extracting is using of multi- and hyperspectral satellites providing data in many spectral bands. Basing on information about radiance intensities a conclusion about cloudiness can be made on per pixel basis. For instance, this approach is widely used for processing of multispectral AVHRR and MODIS data. But three spectral bands of Meteosat do not provide enough information for multispectral cloud recognition algorithms operating on per pixel basis. This causes the need for algorithms, which involves temporal and spatial dependencies in data processing. One of such algorithms is a Markov Random Field segmentation, which allows determining pixel's class with regard to its neighborhoods. Markovian approach allows taking into account different possible distributions of intensities per class and do not introduce global parameters such as thresholds, which is often used in multispectral data processing.

To extract high-level meteorological features fractal approach is used. Meteosat data is modeled by fractional Brownian motion process. Approximation with self-affine fractal allows estimating local fractal dimension of Meteosat image. For this problem Fourier estimator is considered.

Data Preprocessing

Meteosat images come with a lot of noise of two sorts. The first one is the so-called "salt and pepper" noise consisting of noisy pixels uniformly distributed over image. The second one is the impulse burst noise, which distorts images with horizontal streaks of few pixel heights filled with white noise.

In the Space Research Institute NASU-NSAU algorithm for detecting and removing such noise was developed [Phuong, 2004]. On the first step of this algorithm noise streaks is detected and removed by cubic spline interpolation method. During the second step the "salt and pepper" noise is detected by modified median filter and removed by using bit planes approach. The algorithm based on this approach separates an image in 256 planes with binary values. After that, each of these planes is processed separately in order to remove noise.

Cloud Mask Extraction

Following Markovian approach the image is represented as a $n \times m$ matrix of sites S. The neighborhood of site s_{ij} is any subset $\partial_{ij} \subset S$, such that $s_{ij} \notin \partial_{ij}$. With each site s_{ij} two random variables are associated – an intensity X_{ij} (as usual it takes integer value in interval [0; 255]) and a hidden label Y_{ij}. The specific values the random variables take are denoted x_{ij} and y_{ij} respectively. So two sets of variables defined for image S: $X = \{X_{11},...,X_{nm}\}$ and $Y = \{Y_{11},...,Y_{nm}\}$.

Markov Random Fields (MRFs) are widely used for image segmentation [Li, 1995]. With the Hammersley-Clifford theorem the equivalence of MRF and statistical physics Gibbs models was proved [Li, 1995]. This theorem gives us the equation for probability of specific segmentation $P(Y)$

$$P(Y) = \frac{1}{z} e^{\beta V(Y)} = \frac{1}{z} e^{\beta \sum_{i,j} V_{ij}\left(\partial_{ij} \cap \{y_{ij}\}\right)} \tag{1}$$

In this equation, z is normalizing constant necessary for holding the condition $\sum_{Y} P(Y) = 1$. β denotes the image correlation parameter. V is the so-called potential function. Its structure is highly coupled with optimal segmentation of MRF. Defining a particular potential function it is possible to model physics features of segmentation. The right part of equation shows that potential function V can be represented as a sum of potentials defined at each site: $V = \sum_{i,j} V_{ij}$.

For the cloud mask extraction problem the following Markovian model was used: the observed intensity X_{ij} depends only on local label Y_{ij} and a conditional distribution of random variable X_{ij} is Gaussian. The Bayes' theorem about a priory and a posteriori probabilities' relation yields a complete model of intensities and labels coupling [Shiryaev, 1989]:

$$P(Y \mid X) \propto P(X \mid Y)P(Y) = \frac{1}{z} \exp\left\{\sum_{i,j} \beta_{ij} n_{ij}\left(y_{ij}\right)\right\} \times \prod_{i,j} \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left\{-\frac{1}{2\sigma_{ij}^2}\left(x_{ij} - \mu_{ij}\right)^2\right\} \tag{2}$$

Here σ_{ij} and μ_{ij} are a standard deviation and a mean of random variable X_{ij}, n_{ij} is the number of pixels in neighborhood ∂_{ij} with the label equal to Y_{ij}.

The goal of segmentation is to maximize $P(Y \mid X)$ under particular intensities X. This corresponds to obtaining maximum for a posteriori label's estimate:

$$Y : Y^* = \arg\max_{y} \{P(Y \mid X)\}. \tag{3}$$

Results of clouds segmentation and corresponding cloud borders are shown at the fig.1.

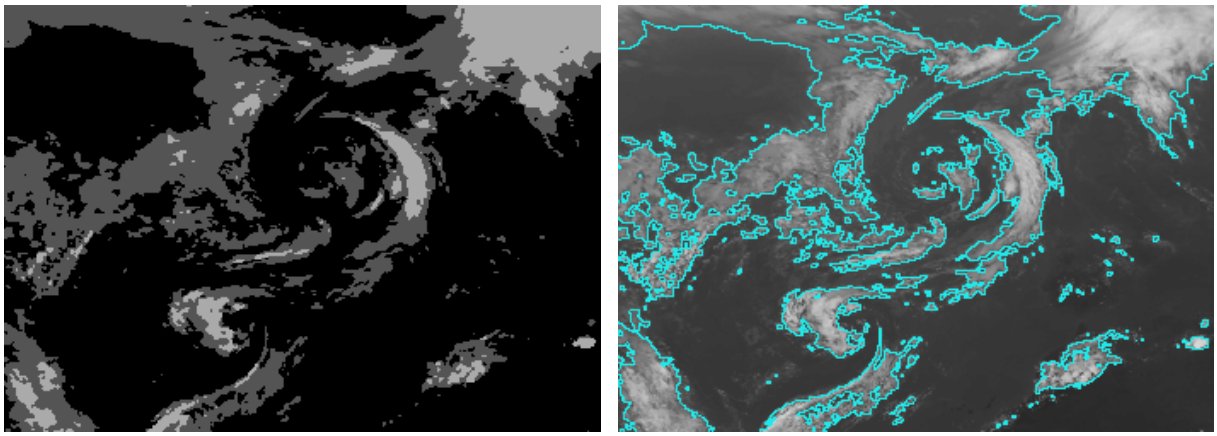


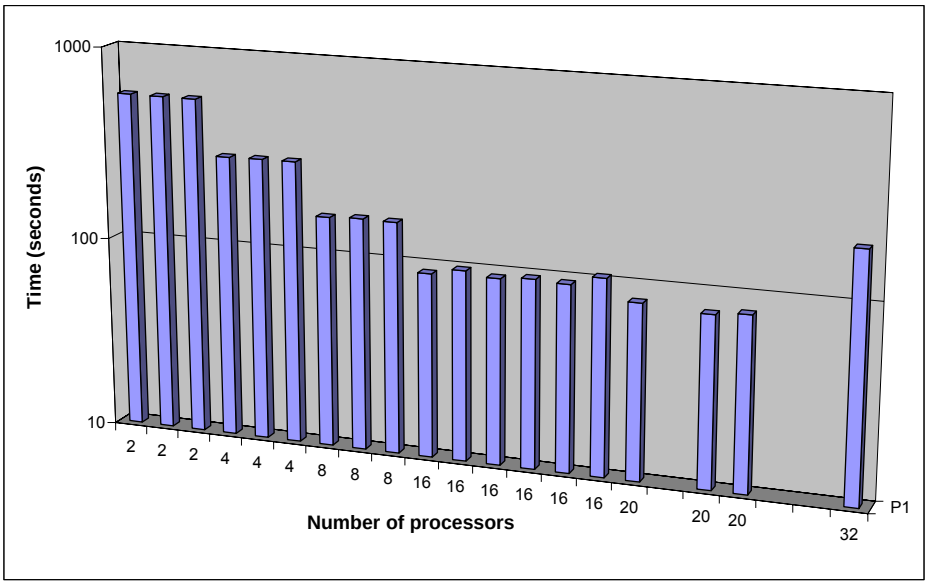
Fig. 1 .MRF segmentation result and corresponding cloud borders.

Parallel Execution Results

High computational complexity of Markovian segmentation algorithm together with large sizes of satellite images determines the need for parallel realization of cloud mask extraction process.

Meteosat image filtering and Markovian segmentation algorithms were implemented using MPI parallel programming interface [MPI, 1997]. Due to locality of dependencies in Markovian image model, it is possible to divide image into almost independent rectangular parts. Then each of these parts is processed by different computational node. Synchronization of several global per-class parameters and image part's borders is performed by means of MPI's group communication functions.

The program was run on the cluster of Institute of Cybernetics NASU consisting of 32 Intel Xeon processors. It has demonstrated good level of parallel acceleration giving almost proportional speed boost with increase of number of computational nodes used (fig. 2). Processing time increasing and productivity slowdown for 32 processors is related with large amount of interprocessors' data transfer, which is fulfilled in sequential way (due to architecture of parallel machine). So according to experimental results the most preferable number of processors for this task is 20. This information is important for load balancing.



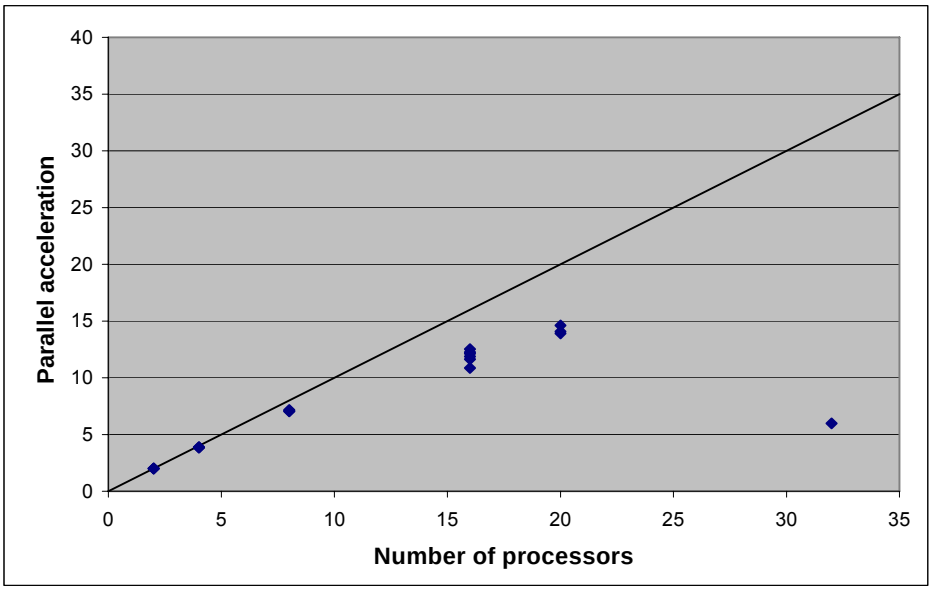


Fig. 2. Performance evaluation of parallel MRF segmentation algorithm.

Fractal Features Extraction

To estimate fractal dimension of Meteosat data water-vapour images are modeled by fractional Brownian motion (FBM) process. Fractional Brownian motion process is the generalization of plain Brownian motion process with expected squared difference in intensity of any two pixels being proportional to the powered distance between the pixels. In the case of two dimensions it is described by the following equation [Potapov, 2002]:

$$\mathbf{E}\left|I(x,y)-I(x+\Delta x,y+\Delta y)\right|^2 \sim \left|\Delta x^2+\Delta y^2\right|^H, \tag{4}$$

where $I(x,y)$ – is the intensity of pixel with coordinates (x,y),

$0<H<1$ – is the Hurst coefficient.

The case of $H=1/2$ corresponds to classical Brownian motion.

It can be shown that two-dimensional FBM process has a Fourier power spectrum

$$\mathbf{E}\left|F(f)\right|^2 \sim 1/f^{\beta}, \tag{5}$$

where the power exponent β, the Hurst coefficient H and fractal dimension D are determined by the following equations:

$$\beta = 2H+2,\ D=3-H \tag{6}$$

Thus knowing β parameter we can estimate the Hurst coefficient H and fractal dimension D.

To examine our algorithm for fractal dimension estimation synthesized images were used. To generate fractional Brownian motion on a two-dimensional grid we use fast Fourier transform filtering. This procedure generates an initial image I_0 as a set of independent Gaussian random variables, $I_0(x,y)\sim N(0,1)$. Then a discrete Fourier transform is applied to image I_0, thus obtaining the grid of Fourier coefficients:

$$F_0(k_x,k_y)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I_0(x,y)\exp\left\{-2\pi i\frac{xk_x+yk_y}{N}\right\},\ k_x,k_y=\overline{1,N} \tag{7}$$

The second step consists in construction of new Fourier coefficients $F_1(k_x, k_y) = F_0(k_x, k_y) \Big/ \left| k_x{}^2 + k_y{}^2 \right|^{\frac{\beta}{4}}$. At last the inverse Fourier transform is applied to $F_1(k_x, k_y)$ coefficients forming result image I_1.

Modeling results for different values of Hurst coefficient H and fractal dimension D from (6) are shown at fig. 3.

a) b) c)

Fig. 3. The examples of FBM generation

a – $H = 0.4$, $D = 2.6$; b – $H = 0.7$, $D = 2.3$; c – $H = 0.9$, $D = 2.1$

Fractal features extraction algorithm based on local Fourier power spectrum investigation. Whole Meteosat image is processed by moving window and corresponding image part used to calculate local Fourier coefficients $F(k_x, k_y)$. According to (5) these coefficients are used to estimate index of power approximation from a linear fit to data $\left\{\left(\log\left|F_1(k_x, k_y)\right|, \log\left|k_x{}^2 + k_y{}^2\right|\right), k_x, k_y = \overline{1, N}\right\}$.

This approach was applied to Meteosat data processing, specifically for fractal dimension detection after cloud mask segmentation. It allows to determine areas of turbulence and to detect sources of some meteorological disasters. Results of satellite data processing are shown at fig. 4.

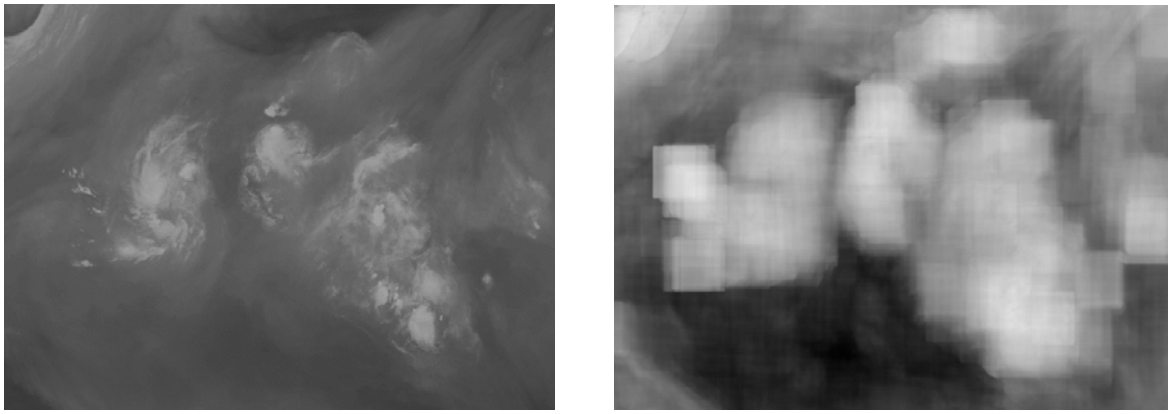


Fig. 4. Original WV image and corresponding fractal dimension estimation.

Conclusions and Further Works

Markovian approach has showed its effectiveness in the task of cloud mask extraction from Meteosat satellite data. Also parallel Markovian segmentation algorithm performed very well exploiting locality of Markovian image model. After cloud mask construction one can implement any other algorithm of higher level satellite data processing. One of them is fractal dimension detection. It allows to determine areas of turbulence and to detect sources of some meteorological disasters.

Further works includes services implementation in GRID environment, which will connect computational cluster and other computational resources with satellite data archives. GRID infrastructure will allow to integrate data processing algorithms with datasets and to provide access to computational tools and there results (products and services) for wide area of users. This kind of investigation is actively carrying out in Space Research Institute of National Academy of Sciences and National Space Agency of Ukraine.

Bibliography

[Phuong, 2004] N. T. Phuong. A Simple and Effective Method for Detection and Deletion of Impulsive Noised Streaks on Images. Problems of Controls and Informatics, 2004. (in Russian)

[Li, 1995] S.Z. Li. Markov Random Field Modeling in Computer Vision. Springer-Verlag, 1995.

[Shiryaev, 1989] A. N. Shiryaev. Probability. Nauka, 1989. (in Russian)

[MPI, 1997] MPI-2: Extensions to the Message-Passing Interface. University of Tennessee, 1997

[Potapov, 2002] A. A. Potapov. Fractals in radiophysics and radiolocation. Logos, 2002. (in Russian)

Authors' Information

Andriy Shelestov, PhD. – senior scientific researcher, Space Research Institute NASU-NSAU, 40 Glushkova St., 03680 Kyiv, Ukraine, e-mail: shelest@dialektika.com

Oleksiy Kravchenko BSc.– system developer, Space Research Institute NASU-NSAU, 40 Glushkova St., 03680 Kyiv, Ukraine, e-mail: akm3000@mail.ru

Michael Korbakov BSc.– system developer, Space Research Institute NASU-NSAU, 40 Glushkova St., 03680 Kyiv, Ukraine, e-mail: rmihael@ukr.net

FORMAL DEFINITION OF ARTIFICIAL INTELLIGENCE [1]

Dimiter Dobrev

***Abstract**: A definition of Artificial Intelligence (AI) was proposed in [1] but this definition was not absolutely formal at least because the word "Human" was used. In this paper we will formalize the definition from [1]. The biggest problem in this definition was that the level of intelligence of AI is compared to the intelligence of a human being. In order to change this we will introduce some parameters to which AI will depend. One of this parameters will be the level of intelligence and we will define one AI to each level of intelligence. We assume that for some level of intelligence the respective AI will be more intelligent than a human being. Nevertheless, we cannot say which is this level because we cannot calculate its exact value.*

***Keywords**: AI Definition, Artificial Intelligence.*

***ACM Classification Keywords**: I.2.0 Artificial Intelligence - Philosophical foundations*

Introduction

The definition in [1] first was published in popular form in [2, 3]. It was stated in one sentence but with many assumptions and explanations which were given before and after this sentence. Here is the definition of AI in one sentence:

AI will be such a program which in an arbitrary world will cope no worse than a human.

[1] This publication is partially supported by the KT-DigiCult-Bg project

From this sentence you can see that we assume that AI is a program. Also, we assume that AI is a step device and that on every step it inputs from outside a portion of information (a letter from finite alphabet Σ) and outputs a portion of information (a letter from a finite alphabet Ω). The third assumption is that AI is in some environment which gives it a portion of information on every step and which receives the output of AI. Also, we assume that this environment will be influenced of the information which AI outputs. This environment can be natural or artificial and we will refer to it as "World".

The ***World*** will be: one set ***S***, one element s_0 of ***S*** and two functions ***World(s, d)*** and ***View(s)***. The set ***S*** contains the internal states of the world and it can be finite or infinite. The element s_0 of ***S*** will be the world's starting state. The function ***World*** will take as arguments the current state of the world and the influence that our device exerts on the world at the current step. As a result, this function will return the new state of the world (which it will obtain on the next step). The function ***View*** gives the information what our device sees. An argument of this function will be the world's current state and the returned value will be the information that the device will receive (at a given step).

Life in one world will be any infinite row of the type: $d_1, v_1, d_2, v_2, \ldots$ where v_i are letters from Σ and d_i are letters from Ω. Also, there has to exist infinite row $s_0, s_1, s_2, \ldots$ such that s_0 is the staring state of the world and $\forall i>0\ v_i=View(s_i)$ and $\forall i\ s_{i+1}=World(s_i, d_{i+1})$. It is obvious that if the world is given then the life depends only on the actions of AI (i.e. depends only on the row $d_1, d_2, d_3, \ldots$).

In order to transform the definition in [1] and to make it formal, we have to define what is a program, what is a good world and when one life is better than another.

The first task is easy because this work is done by Turing in the main part. Anyway, the Turing definition of program is for a program which represents function, but here we need a transducer which inputs the row $v_1, v_2, v_3, \ldots$ and outputs the row $d_1, d_2, d_3, \ldots$. So, we will make a modification of the definition of Turing machine [9].

Our second task is to say what is a good world. It was written in [1] that if you can make a fatal error in one world then this world is not good. What is world without fatal errors needs additional formalization.

The next problem is to say when one life is better than another. This is done in [1] but there are some problems connected with the infinity which have to be fixed.

The last task is to say how intelligent our program should be and this cannot be done by comparison with a human being.

What is a Program

We will define a program as a Turing machine [9]. Let its alphabet Δ consist of the letters from Σ, from Ω, from one blank symbol λ and from some service signs.

Let our Turing machine have finite set of internal states ***P***, one starting state p_0 and a partial function $F : P \times \Delta \rightarrow P \times \Delta \times \{Left, Right\}$.

The Turing machine (TM) is a step device and it makes steps in order to do calculations. On the other hand, AI is a step device and its life consists of steps. In order to make distinction between these two types of steps we will call them small and big steps. When we speak about time we will mean the number of big steps.

Of course, our TM will start from the state p_0 with infinite tape filled with the blank symbol λ. How our TM will make one small step. If it is in state ***p*** and if its head looks at the letter δ then $F(p, \delta)$ will be a 3-tuple which first element is the new state after the small step, the second element will be the new letter which will replace δ on the tape and the third element will be direction in which the head will move.

How will our TM (which is also our AI) make one big step? This will happen when after a small step the new state of TM is again p_0. At this moment our TM has to output one letter d_i and to input one letter v_i. We will assume that the letter which is outputted is that which is written on the place of δ on this small step. But how after outputting the letter d_i will our TM input the letter v_i? We will put this letter on the place where the head after the small step is. In this way we are intervening in the work of the TM by replacing one symbol from the tape with another. The replaced symbol is lost in some sense because it will not influence the execution of the TM from this small step on.

We will assume that our TM is outputting only letters from Ω (no letters from the rest of Δ). Also, we assume that our TM never hangs. TM hangs if after reading some input $v_1, v_2, \ldots, v_n$ it stops because it falls into some state p and its head falls on some letter δ such that $F(p, \delta)$ is not defined. TM also hangs if after reading of some input $v_1, v_2, \ldots, v_n$ it makes infinitely many small steps without reaching the state p_0 (without making of big steps anymore).

After this we have a formal definition of a program. We have to mention that there is no restriction on the number of the small steps which TM needs to make for one big step. This number has to be finite but it is not restricted. Maybe it is a good idea to add one parameter ***Max_number_of_small_steps_in_AI*** in order to exclude some decisions for AI which are combinatory explosions. (If we restrict the number of small steps then we have to restrict also the number of service signs in Δ because we can speed up the TM by increasing the size of its alphabet.) If we want to use AI as a real program on a real computer then we have to take into consideration that the memory of the real computers is limited. So, we can restrict also the size of the tape. Anyway, we will not care about the efficiency of AI and we will not make such restrictions.

What is a World without Fatal Errors

It is very difficult to define what is a world without fatal errors. That is why we will do something else. We will restrict our set of worlds is such a way that the new set will contain only worlds without fatal errors.

Let our world look like one infinite sequence of games. Let every game be independent from the previous ones. Let us have three special letters in Σ, which we will call final letters. Let this letters be ***{victory, loss, draw}***. Let every game finish with one of the final letters. Let every game be shorter than 1000 big steps.

Remark 1: Our definition of AI will depend on many parameters. In order to simplify the exposition we will fix these parameters to concrete numbers. Such parameter is the maximum number of steps in a game which will be fixed to 1000. Also, in order to simplify the exposition we will use different numbers for different parameters.

Remark 2: The only parameters in our definition which are not numbers are the alphabets Σ and Ω. We will assume that these alphabets are fixed and that Ω has at least 2 letters and Σ has at least 2 letters which are not final. (If Ω has only one letter then there will be no choice for the action of AI and the world will be independent from this action. If Σ has only one letter, which is not final, then the game will be blind because AI will not receive any information until the end of the game. Therefore, the minimum for $|\Sigma|$ is 5.)

We will assume that the world has three special internal states ***{s_victory, s_loss, s_draw}***, which we will call final states. Let these states be indistinguishable from the state s_0 for the function ***World***. This means that the world will behave in the final states in the same way as if it was in the starting state. Let the function ***View*** distinguish the final states and return from them the letters ***victory***, ***loss*** and ***draw*** respectively. Also, the final states will be the only states on which the function ***View*** will return one of the letters ***{victory, loss, draw}***.

After the restriction of the definition of ***World***, we can be sure that there are no fatal errors in our world because the life in such a world is an infinite sequence of games and if we lose some games (finitely many) then this will not be fatal because every new game is independent from the previous ones. Also, we are sure that a new game will come sooner or later because every game is finite (i.e. previous game is shorter than 1000 steps).

When is One Life better than Another

In [1] we gave the following definition for the meaning of the life: One life is better than another if it includes more good letters and fewer bad letters. Here good letters will be ***{victory, draw}*** and bad letters will be ***{loss, draw}***. So, here life is good if we win often and lose seldom.

We want to introduce one function ***Success*** which will evaluate with a real number every life in order to say how good it is. For that we will define first the function ***Success*** for the every beginning of life (all beginnings are finite). After that we will calculate the limit of ***Success*** when the size of the beginnings goes to infinity and this limit will be the value of ***Success*** for the entire life.

The function ***Success*** can be defined for the beginnings like the difference between the number of victories and the number of losses. This is not a good idea because then the limit of ***Success*** will possibly converge to infinity (plus or minus infinity). It is a better idea to calculate the percentage of victories. So, we define ***Success***

as ***(2.N_victory +N_draw)/ (2.N_games)***. Here ***N_victory*** is the number of victories (analogically for ***N_draw*** and ***N_games***). Function ***Success*** will give us a number between 0 and 1 for every beginning and its limit will be also between 0 and 1. The only problem is that ***Success*** may not have a limit. In such a case we will use the average between limit inferior and limit superior.

Trivial Decisions

Now we have a really formal definition of AI and this gives us the first trivial decision for AI.

TD1 will be the program which plays at random until the first victory. After that TD1 repeats this victory forever. For this TD1 needs only to remember what it did in the last game. If the last game was victorious then it can repeat this last game because the function World is deterministic and if TD1 is doing the same then the world will do the same too.

TD1 is perfect in all worlds in which the victory is possible. If the victory is not possible then TD1 will play at random forever. That is why we will make TD2 which will be perfect in all worlds.

TD2 will be this program which tries sequentially all possible game's strategies until it finds the first victory and after that repeats this victory forever. If there is no victorious game strategy then TD2 repeats the last draw game forever. If the draw game is not possible too then TD2 plays at random. (It is important that the game's length is not more than 1000. This means that the number of the game's strategies is finite.)

TD2 is perfect in all worlds and this means that it is a trivial decision on our definition for AI. Really, the definition stated that AI has to cope no worse than a human but for the perfect program this is true because it copes no worse than anything even no worse than a human being.

It is suspicious that such simple program like TD2 can satisfy our definition for AI. That is why we will change the definition by accepting more possible worlds. It is too restrictive to assume that the game is deterministic and every time you do the same the same will happen.

Nondeterministic Games

We will assume that the function ***World*** is not deterministic. It is better to say that it is multy-valued function, which chooses at random one of its possible values. Let every possible value correspond to one real number, which is the possibility for this value to be chosen. We will assume also that $\forall s\ \forall \omega$ ***World(s, ω)*** has at least one value and that $\forall s\ \forall \omega$ ***(*** for every two different values of ***World(s, ω)*** the function ***View*** returns different result***)***.

Remark 3: The latter means that if something nondeterministic happens this information will be given to AI immediately by the function ***View***. There is no sense to assume existence of a nondeterministic change which cannot be detected immediately but later or even which cannot be detected never.

Now we will ask the question what is the best strategy in such a world and we will offer a program, which will be almost perfect. Before that we need several definitions:

Definition 1: ***Tree of any game***. It will have two types of vertices. The root and the other vertices which are on even depth will be the vertices of type AI (because they correspond to the moments when AI has to do its choice). The vertices which are on odd depth will be vertices of the type world (because they correspond to the moments when the world will answer at random). From the vertices of type AI go out $|\Omega|$ arcs and to every such arc corresponds one of the letters from Ω. There is one exception. If the arc which is right before this vertex corresponds to a final letter, then this vertex is a leaf. From the vertices of type world go out $|\Sigma|$ arcs and to every such arc corresponds one of the letters from Σ. Here there is an exception again. If this vertex is on depth 1999, then only three arcs go out and these three arcs correspond to the final letters.

You can see that the tree of any game is finite and its maximum depth is 2000 (because games are not longer than 1000 steps). Nevertheless, there are leaves on any even depth between 2 and 2000.

Definition 2: ***Tree of any 100 games***. Let us take the tree of any game. Let us replace all of its leaves with the tree on any game. Let us repeat this operation 99 times. The result will be the tree of any 100 games (which is 100 times deeper than the tree of any game).

From the ***tree of any game*** we will receive ***Strategy for any game***. This will be its subtree which is obtained by choosing one vertex from the successors of every vertex of the type AI and deleting the rest successors

(and their subtrees). Analogically we make ***Strategy for any 100 games*** like a subtree from the ***tree of any 100 games***. We have to mention that the ***Strategy for any 100 games*** can be different from repeating one ***Strategy for any game*** 100 times. The reason is because the strategy on the next game can depend on the previous games.

Definition 3: ***Tree of this game.*** For every concrete game (i.e. concrete world) we can construct the tree of this game as a subtree from the tree of any game. We will juxtapose internal states of the world to the vertices of type AI in the time of this construction. First, we will juxtapose the state s_0 to the root. Let k_0, k_1 and k_2 be vertices and let k_1 be successor of k_0 and k_2 be successor of k_1. Let k_0 be vertex of type AI and let the state s be juxtaposed to it. Let the letters ω and ε be juxtaposed to the arcs $< k_0, k_1 >$ and $< k_1, k_2 >$. In this case if $\varepsilon \neq View(World(s, \omega))$ for every value of $World(s, \omega)$ then we delete the vertex k_2 (and its subtree). In the opposite case we juxtapose k_2 to this value of $World(s, \omega)$ for which $\varepsilon = View(World(s, \omega))$. This value is only (look at remark 3). Also, we will juxtapose the possibility ε to be the value of $View(World(s, \omega))$ to the arc $< k_1, k_2 >$. So, one letter and one possibility will be juxtaposed to the arc $< k_1, k_2 >$.

Analogically to the strategy for any game we can make strategy for this game. We have to say that if the World is deterministic (i.e. every vertex of type world has only one successor) then the strategy for this game is a path (a tree without branches). In this case the paths in the tree of this game are exactly the strategies for this game. This was used from TD2 in order to try all strategies.

Max-Sum Algorithm

For every vertex of the tree of this game we can calculate the best possible success (this is our expectation for success, if we play with the best strategy from that vertex on).

1. The best possible success for the leaves will be 1, 0 and 1/2 for the states ***s_victory***, ***s_loss*** and ***s_draw*** respectively.

2. If the vertex is of type AI, then its best possible success will be the maximum from the best possible successes of its successors.

3. If the vertex is of type world, then its best possible success will be the sum $\sum Possibility(i).BestPossibleSuccess(i)$. Here i runs through all successors of this vertex.

The algorithm for calculating the best possible success can also be used to calculate the best strategy in this game (the best strategy can be more than one). This algorithm looks like the Min-Max algorithm, which we use in chess. Anyway, this is different algorithm, to which we will refer as Max-Sum algorithm. The difference is essential because in the chess we assume that we play against someone who will do the worst thing to us (remark 4). Anyway, in the arbitrary world we cannot assume that the world is against us. For example, when you go to work you go first to the parking lot in order to take your car. If your car is stolen, then you go to the bus stop in order to take the bus. If every time you were presumed the worst case, then you would go directly to the bus stop.

New Trivial Decisions

Now we can calculate the best possible success for any game and we will give the next trivial decision (TD3), which will do the best in every game. This means that the success of TD3 for one world will be equal to its best possible success.

TD3 will be the program which plays at random for long time enough. In this time TD3 collects statistical information for the ***tree of this game*** and builds inside its memory this tree together with the values of all possibilities. After that time TD3 starts playing by the use of Max-Sum algorithm.

TD3 gives the perfect decision in any world but TD3 is impossible because we cannot say when enough statistical information is selected. Anyway, possible is something which is a little bit worse. For every $\varepsilon > 0$ we will make TD4, which for every world will make success on a distance no more than ε from the best possible.

TD4 will be this program which simultaneously collects statistical information for the ***tree of this game*** and in the same time plays by the use of Max-Sum algorithm on the base of statistics, which is collected up to the current moment. In order to collect statistics TD4 makes experiments which contradict to the recommendations of

Max-Sum algorithm. Such experiments are made rarely enough to have success on a distance not bigger than ε from the best possible success.

We can choose the value of ε to be as small as we want. Anyway, the price for the small value of ε is the longer time for education (because of rare experiments). We will call the parameter ε "courage". Here we receive a surprising conclusion that if AI is more cowardly it is closer to perfection (this is true only in the case of infinite life).

TD4 is a decision for our definition of AI because it is only on ε distance to perfection unlike the people who are much farther from perfection. We have to mention that in some sense TD4 is not as trivial as TD2, because TD4 represents awful combinatory explosion in the execution time (number of small steps) and in the memory size. Anyway, we said that we will not care about the efficiency of AI for the moment. On the other hand, there is one additional problem, which is present in both TD2 and TD4, and which makes them both useless. This is the problem of the combinatory explosion of the educational time. Imagine that you are playing chess at random against deterministic partner. How long will you need to make accidental victory? Or in case your partner is not deterministic. How long will you need to play all possible game's strategies and try each one several times in order to collect statistical information on how your partner reacts in every case?

Finite Life

In some sense TD2 and TD4 are extremely stupid because they need extremely long time for education. Really, educational time and level of intelligence are two different parameters of the mind and if one of them is better then this can be at the expense of the other. For example, a human being needs about a year to learn to walk, which is much worse in comparison to most animals. Some of the greatest scientists had bad results in school, which can be interpreted as a fact that they advanced slower than the ordinary people.

Therefore, the educational time is important and it has to be limited in order to make our definition useful. This will be done by changing the life length from infinite to finite. We will assume that the length of the life is 100 games. Each game has maximum 1000 steps, which means that the life length is not bigger than 10,000 steps. Now the success of the life will not be the limit of the ***Success*** function but the value of this function for the first 100 games.

After this we can look for program which makes a good success in an arbitrary world, but this is not a good idea because the arbitrary world is too unpredictable. Human beings use the assumption that the world is simple and that is why they cope very well in a more simple environment and they are totally confused if the environment is too complicated. Therefore, we have to restrict the complexity of the world and give bigger importance to the more simple worlds. For this restriction we will use Kolmogorov Complexity [8]. The parameter which restricts the complexity of the world will be the level of intelligence of AI.

Kolmogorov Complexity

First we need a definition of program which calculates the functions World and View. For this we will use the same definition of TM as for the program which was our AI. There will be some small differences: The alphabet of the Turing Machine of the world (TM_W) will be $\Sigma \cup \Omega \cup \{\lambda\}$ (the only service symbol will be λ). Also, TM_W will input the row $d_1, d_2, d_3, \ldots$ and output the row $v_1, v_2, v_3, \ldots$. At the beginning TM_W will start with tape on which d_1 is at the head position and the rest is λ. At the end of the first big step TM_W will output v_1 and input d_2. F will be set of 5-tuples which is a subset to $P \times \Delta \times P \times \Delta \times \{Left, Right\}$. This means that F is not a function but a relation (because it will represent multy-valued function). We will assume that $\forall s \ \forall \delta \ \exists 5\text{-tuple} \in F$ whose first two elements are s and δ (this makes the reasons for hanging with one less). The 5-tuples in F whose third element is p_0 will be called output 5-tuples. The fourth element of output 5-tuples has to be letter from Σ (this is not necessary but sufficient condition in order TM_W to output only letters from Σ). We will allow nondeterministic behavior only for output 5-tuples. This means that if two different 5-tuples have the same first and second elements then they both have to be output 5-tuples. There will be no two 5-tuples which differ only at the fifth element (we cannot have a choice between two nondeterministic 5-tuples which output the same letter - look again at remark 3). It will be more interesting if we assume that nondeterministic 5-tuples have additional

parameter which shows the possibility for each of them to be chosen. Nevertheless, we will assume that this possibility is distributed equally and that we do not have such additional parameter.

According to this definition, the internal states of the world will be the states of the tape of the TM_W. If we want to have worlds without fatal errors we have to clean the tape of TM_W after each game (after printing any final letter). Nevertheless, we will not do this because the absence of fatal errors was important when we had infinite life and when we counted on that sooner or later all errors will be compensated. For real AI is better to assume some connection between games. Otherwise AI will not remember what was done in the last game or it will remember it but it will not know whether this was in the last game or in some other game from the previous ones.

Another question is what we will do with TM_W which hangs. We do not want to exclude these programs from our definition (at least because we cannot check this characteristic). That is why we will assume that if one TM_W makes more than 800 small steps without making a big step then we will interrupt it with output "draw". This means that it will do the next small step in the same way as if the 5-tuple executed at this moment had third element p_0 and fourth element "draw". Also, if one TM_W makes 1000 big steps without outputting any final symbol then the output of the next big step will be "draw". We need this in order to keep the games finite, which is important in order to keep the life finite (the life is 100 games).

We will define the size of TM_W as the number of internal states plus the level of indefiniteness (this is the minimal number of nondeterministic 5-tuples, which have to be deleted from F in order to make it deterministic or this is the number of all nondeterministic 5-tuples minus the number of different groups of nondeterministic 5-tuples).

So, we will restrict the set of possible worlds to those generated by Turing Machine whose size is not bigger than 20. The maximum size of the TM_W will be the level of intelligence of AI. The simpler worlds will be more important because they are generated from more than one TM_W and that is why we will count their result more than once.

Remark 4: It looks like that two Turing machines (the world and AI) play against each other. Anyway, this is wrong because the world does not care about AI and it dose not play against AI.

Final Definition of AI

Now everything is fixed. We have finite lives, which are exactly 100 games. We had selected the success function that will evaluate these lives. Also, we made a finite set of worlds which consist of the worlds generated from the TM_W with size not bigger than 20. Now we can define AI as this program which will make the best average success in the selected worlds. Such program exists and it will be the next trivial decision (TD5).

The number of all strategies for playing 100 consecutive games is finite. The number of selected worlds is also finite. We can calculate the expected success of any strategy in any world. The average success of a strategy will be the arithmetical mean from its expected success in any world. (The calculation of the expected success of a strategy in a world is easy if the world is deterministic. In this case, we will have simply to play 100 games with this strategy in this world. In the opposite case, if the world is nondeterministic then we have to use Max-Sum algorithm, which is theoretically possible, but in practice it is a combinatory explosion. Nevertheless, even if the worlds were only deterministic, we would have combinatory explosion again from the number of worlds and from the number of strategies.)

Hence, TD5 will be this program which calculates and plays the best strategy (this which average success is biggest). Such program is easy to be written but it is very difficult to wait until it makes its first step. The situation with the perfect chess playing program is analogical. (It plays chess by calculating all possible moves.) This program can be written very easy but the time until the end of the universe will be not enough for it to make its first move.

It will be too restrictive if we define AI as the best program (such as TD5 or such as any other program equivalent to TD5). It will be better if we say that AI is a program whose average success is not more than 10% from the best (from TD5). Such definition is theoretically possible, but practically inconvenient. The reason for this is the fact that the value of the average success of TD5 can be theoretically calculated, but in practice this is absolutely impossible. So, if we select such definition we will not be able to check it for a concrete program.

Final definition:

AI will be a program which makes more than 70% average success in the selected set of worlds.

Assumption 1: Here we assume that the average success of TD5 is about 80%. If this conjecture is true then there exists a program which satisfies the definition (at least TD5 do). If the average success of TD5 is smaller than 70% then there is no such a program (of course, in such case we can change this parameter and make it smaller than 70%).

The advantage of this definition of AI is that we can check it for a concrete program. Of course, we cannot calculate the average success for any program due to the combinatory explosion, but we can calculate it approximately by the methods of the statistics. For this, we will select at random N worlds (world is TM_W with size not bigger than 20) and we will play 100 consecutive games in every world. If the world is deterministic then this will give us the expected success of our program in this world. If it is not deterministic then we will play at random in this world. This will give us statistically good evaluation of the expected success because the possibility to be extremely lucky in 100 games is very small (so is the possibility to be extremely unlucky). If N (the number of the tested worlds) is big then the statistical result will be close to the average success of our program.

If $|\Sigma \cup \Omega \cup \{\lambda\}| = 5$ (which is the minimum - remark 2) then the number of deterministic TM_W with 20 states is 200 on power of 100. If we take the number of nondeterministic TM_W with 19 states and level of indefiniteness one (which means with two nondeterministic 5-tuples) then this number is many times smaller than 200 on power of 100. In order to use the method of statistics we have to calculate how many times is this number smaller. Otherwise we will use wrong correlation between deterministic and nondeterministic TM_W. Anyway, such wrong correlation will make an unessential change in the definition of AI.

Conclusion

Now we have definition of AI and at least one program (TD5) which satisfies it (with assumption 1). The first question is: Does this definition satisfy our intuitive idea that AI is a program which is more intelligent than a human being. Yes, but for some values of the parameters educational time and level of intelligence. In this paper the educational time was fixed on 100 games each of them no longer than 1000 steps (educational time is equal to the life length because we learn all our life). The level of intelligence here was fixed on 20. Which means that we assume that we can find a model of the world which is TM_W with size not bigger than 20. We cannot say what is the exact level of intelligence of the human being.

The second question is: Is TD5 which satisfies the definition the program which we are looking for. The answer is definitely no. We are looking for a program which can work in real time. Also, our intuitive idea is that AI should build a model of the world and on the base of it AI should plan its behavior. Look at [6, 7]. Instead of this, TD5 uses brutal force in order to find the best strategy. Even TD5 will not know what to do on the game 101 because its strategy is only for 100 games.

Here we will offer the last trivial decision (TD6), which corresponds better to our intuitive idea for AI.

Let TD6 be the program which tries all deterministic TM_W and accepts as a model of the world the first one (the shortest one) which generates the life until the present moment. After selecting a model (which will be a big problem for more complicated worlds) TD6 will use this model and the Max-Sum algorithm in order to plan its next move. Here Max-Sum will not calculate until the end of the life or even until the end of the game because this will give a combinatory explosion. Instead, it will calculate several steps like chess playing programs do.

You can find a program similar to TD6 in [5]. Really, in [5] we are looking for TM_W, which is a generator of infinite sequence $v_1, v_2, v_3, \ldots$ instead of looking for transducer from $d_1, d_2, d_3, \ldots$ to $v_1, v_2, v_3, \ldots$. Also, [5] does not make moves (there is no Max-Sum algorithm for calculating the next move). The only thing which [5] does is to predict the next number of the sequence. Anyway, in [5] you can see that searching for a model in the set of all TM_W works only if the model is very simple. If the size of TM_W is bigger than 5, the result is combinatory explosion.

If we modify TD6 in order to accept also the nondeterministic Turing machines as models of the world then we will have too much possible models. In this case we have to consider more than one model and to evaluate each possible model in order to see how reliable it is.

Anyway, TD6 and its modifications are not the program which we are looking for, although TD6 can be done to satisfy the definition (because the definition does not say anything about the efficiency of AI). The program which we are looking for is much closer to that one which is described in [6, 7]. The problem in TD6 is that it looks for a model of the world which consists from only one item. It is better if the model is a set of many items (the items can be Turing machines, final automata or logical formulas). When we make a theory in logic then it consist from a set of axioms and we can change smoothly the theory by modifying, adding or deleting one axiom. Any theory in logic is a model of some world. AI has to use similar models which can be modified smoothly.

Bibliography

[1] Dobrev D. D. A Definition of Artificial Intelligence. In: Mathematica Balkanica, New Series, Vol. 19, 2005, Fasc. 1-2, pp.67-74.

[2] Dobrev D. D. AI - What is this. In: PC Magazine - Bulgaria, November'2000, pp.12-13 (in Bulgarian, also in [4] in English).

[3] Dobrev D. D. AI - How does it cope in an arbitrary world. In: PC Magazine - Bulgaria, February'2001, pp.12-13 (in Bulgarian, also in [4] in English).

[4] Dobrev D. D. AI Project, http://www.dobrev.com/AI

[5] Dobrev D. D. First and oldest application, http://www.dobrev.com/AI/first.html (1993)

[6] Dobrev D. D. Testing AI in one Artificial World. Proceedings of XI International Conference "Knowledge-Dialogue-Solution", June 2005, Varna, Bulgaria, Vol.2, pp.461-464.

[7] Dobrev D. D. AI in Arbitrary World. Proceedings of the 5th Panhellenic Logic Symposium, July 2005, University of Athens, Athens, Greece, pp.62-67.

[8] Kolmogorov A. N. and Uspensky V. A. Algorithms and randomness. - SIAM J. Theory of Probability and Its Applications, vol. 32 (1987), pp.389-412.

[9] Turing, A. M. On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, Series 2, 42, 1936-37, pp.230-265.

Author's Information

Dimiter Dobrev - Institute of Mathematics and Informatics, BAS, Acad.G.Bonthev St., bl.8, Sofia-1113, Bulgaria; P.O.Box: 1274, Sofia-1000, Bulgaria; e-mail: d@dobrev.com

PROGRAMMING PARADIGMS IN COMPUTER SCIENCE EDUCATION

Elena Bolshakova

***Abstract**: Main styles, or paradigms of programming – imperative, functional, logic, and object-oriented – are shortly described and compared, and corresponding programming techniques are outlined. Programming languages are classified in accordance with the main style and techniques supported. It is argued that profound education in computer science should include learning base programming techniques of all main programming paradigms.*

***Keywords:** programming styles, paradigms of programming, programming techniques, integration of programming techniques, learning programming paradigms*

***ACM Classification Keywords**: D.1 Programming Techniques – Functional Programming, Logic Programming, Object-oriented Programming*

Introduction

Several main ***styles*** (or ***paradigms***, or models) ***of programming*** – imperative, functional, logic and object-oriented ones – were developed during more than forty-year history of programming. Each of them is based on specific algorithmic abstractions of data, operations, and control and presents a specific mode of thinking about program and its execution. Various ***programming techniques*** (including data structures and control mechanisms) were elaborated rather independently within each style, thereby forming different scopes of their applicability. For instance, the object-oriented style and corresponding techniques are suitable for creating programs with complicated data and interface, while the logic style is convenient to program logic inference.

Though modern programming languages [Finkel, 1996] usually include programming techniques from different styles, they may be classified according to the main style and techniques supported (e.g., programming language Lisp is a functional language while it includes some imperative programming constructs).

Nowadays, for implementation of large programming project, techniques from different paradigms are required, mainly because of complexity and heterogeneity of problems under solution. Some of them are problems of complex symbolic data processing, for which programming techniques of functional and logic languages (e.g., Lisp [Steele, 1990] or Prolog [Clocksin, 1984]) are adequate. The other problems can be easily resolved by means of popular imperative object-oriented languages, such as C++ [Stroustrup, 1997].

Below we explain our point that acquirement of programming techniques of all main paradigms belongs to background knowledge in the field of computer science. Accordingly, learning of modern programming languages should be complemented and deepened by learning of programming paradigms and their base techniques.

Programming Paradigms

The ***imperative*** (procedural) programming paradigm is the oldest and the most traditional one. It has grown from machine and assembler languages, whose main features reflect the John von Neuman's principles of computer architecture. An imperative program consists of explicit commands (instructions) and calls of procedures (subroutines) to be consequently executed; they carry out operations on data and modify the values of program variables (by means of assignment statements), as well as external environment. Within this paradigm variables are considered as containers for data similar to memory cells of computer memory.

The ***functional*** paradigm is in fact an old style too, since it has arisen from evaluation of algebraic formulae, and its elements were used in first imperative algorithmic languages such as Fortran. Pure functional program is a collection of mutually related (and possibly recursive) functions. Each function is an expression for computing a value and is defined as a composition of standard (built-in) functions. Execution of functional program is simply application of all functions to their arguments and thereby computation of their values.

Within the ***logic*** paradigm, program is thought of as a set of logic formulae: axioms (facts and rules) describing properties of certain objects, and a theorem to be proved. Program execution is a process of logic proving (inference) of the theorem through constructing the objects with the described properties.

The essential difference between these three paradigms concerns not only the concept of program and its execution, but also the concept of program variable. In contrast with imperative programs, there are neither explicit assignment statements nor side effects in pure functional and logic programs. Variables in such a program are similar to those in mathematics: they denote actual values of function arguments or denote objects constructed during the inference. This peculiarity explains why functional and logic paradigms are considered as non-traditional.

Within the ***object-oriented*** paradigm, a program describes the structure and behavior of so called objects and classes of objects. An object encapsulates passive data and active operations on these data: it has a storage fixing its state (structure) and a set of methods (operations on the storage) describing behavior of the object. Classes represent sets of objects with the same structure and the same behavior. Generally, descriptions of classes compose an inheritance hierarchy including polymorphism of operations. Execution of an object-oriented program is regarded as exchange of messages between objects, modifying their states.

Table 1. Features of the main programming paradigms

Paradigm	*Key concept*	*Program*	*Program execution*	*Result*
Imperative	Command (instruction)	Sequence of commands	Execution of commands	Final state of computer memory
Functional	Function	Collection of functions	Evaluation of functions	Value of the main function
Logic	Predicate	Logic formulas: axioms and a theorem	Logic proving of the theorem	Failure or Success of proving
Object-oriented	Object	Collection of classes of objects	Exchange of messages between the objects	Final state of the objects' states

The object-oriented paradigm is the most abstract, as it's basic ideas can be easily combined with the principles and programming techniques of the other styles. Really, an object method may be interpreted as a procedure or a function, whereas sending of message as a call of procedure or function. Contrarily, traditional imperative paradigm and non-traditional functional and logic ones are poorly integrated because of their essential difference. Distinguishing features of the main programming paradigms are clarified in Table 1.

Programming Languages and Programming Techniques

Each algorithmic language was initially evolved within a particular paradigm, but later it usually accumulates elements of programming techniques from the other styles and languages (genesis of languages and relations between them are shown in Fig.1).

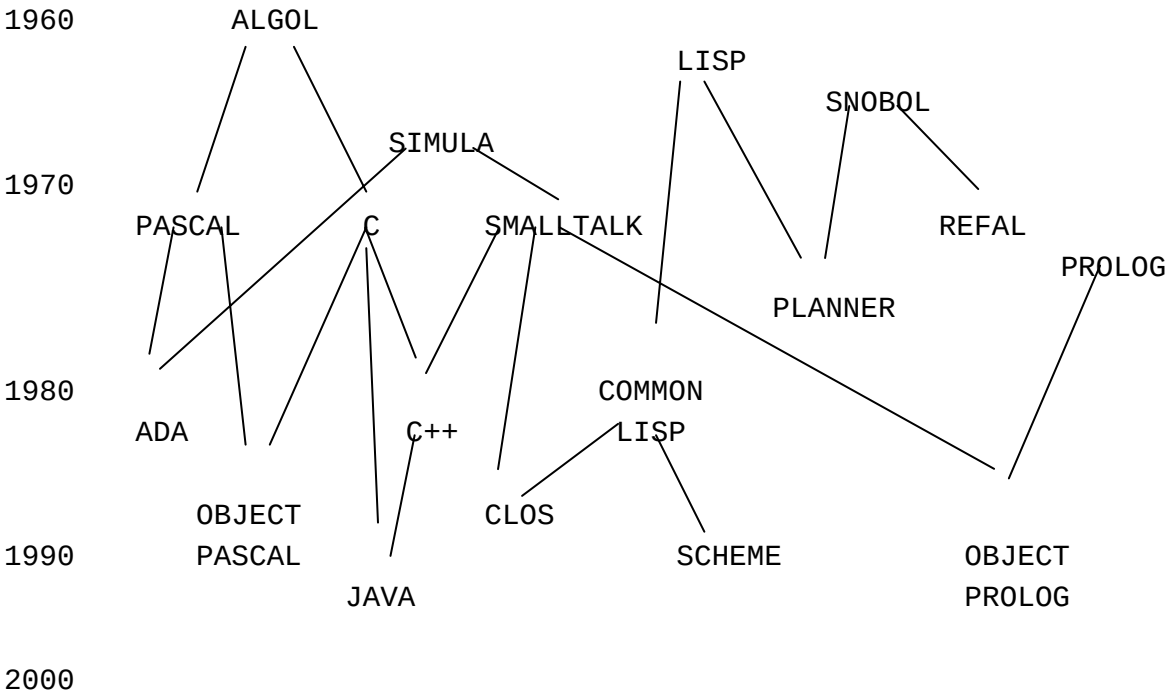


Fig. 1. Genealogy of programming languages

Hence, as a rule, most languages include a kernel comprising programming techniques of one paradigm and also some techniques from the other paradigms. We can classify languages according to paradigms of their kernels. The following is a classification of several famous languages against the main paradigms:

- Imperative paradigm: Algol, Pascal, C, Ada;
- Functional paradigm: Lisp, Refal, Planner, Scheme;
- Logic paradigm: Prolog;
- Object-oriented paradigm: Smalltalk, Eiffel.We could notice that Smalltalk [Goldberg, 1982], the first object-oriented language, is not popular because of complexity of its syntax and dynamic semantics. But its basic object ideas (abstraction of object's state and behavior, encapsulation and inheritance of state and behavior, polymorphism of operations) are easily integrated with the principles of programming languages of the other styles. For this reason, the object-oriented paradigm became widespread as soon as it was combined with traditional imperative paradigm. To be more precise, it became widespread when it was embedded into the popular imperative languages C and Pascal, thereby giving imperative object-oriented languages C++ and Object Pascal.

Analogous integration of object-oriented principles with programming techniques of the other paradigms has led to object-oriented variants of non-traditional languages. For example, the language Clos is an object-oriented Lisp developed on the base of Common Lisp [Steele, 1990], the popular version of Lisp. Modern programming languages, which are combinations of two paradigms, are:

- Imperative + Object-oriented paradigms: C++, Object Pascal, Ada-95, Java;
- Functional + Objects-oriented paradigms: Clos;
- Logic + Object-oriented paradigms: Object Prolog.

Programming techniques elaborated within the traditional imperative paradigm and imperative languages, are well known [Finkel, 1996]: control structures include cyclic and conditional statements, procedures and functions, whereas data structures comprise scalar and compound types – arrays, strings, files, records, etc. Programming techniques of imperative object-oriented languages also includes object types and corresponding techniques, such as virtual procedures and functions.

Programming languages based on non-traditional paradigms provide rather different data types and control structures [Field, 1988], they also differ from traditional languages in the mode of execution: interpretation instead of compilation applied for imperative and imperative object-oriented languages.

Unlike imperative languages, logic and functional languages are usually recursive and interpretive, and most of them are oriented towards symbolic processing. Besides recursion, programming techniques developed within these languages include:

- flexible data structures for representing complex symbolic data, such as list (Lisp) or term (Prolog);
- pattern matching facilities (Refal, Prolog) and automatic backtracking (Planner, Prolog);
- functionals, i.e. high order functions (Lisp, Scheme);
- mechanism of partial evaluations (Refal).

Programming techniques elaborated within corresponding programming style and programming languages have its own scope of adequate applications. Functional programming is preferable for symbolic processing, while logic programming is useful for deductive databases and expert systems, but both of them are not suitable for interactive tasks or event-driving applications. Imperative languages are equally convenient for numeric and symbolic computations, giving up to most of the functional languages and Prolog in the power of symbolic processing techniques. The object paradigm is useful for creating large programs (especially interactive) with complicated behavior and with various types of data.

Integration of Programming Techniques

Nowadays, imperative object-oriented languages C++, Java, and Object Pascal supported by a large number of developing tools are the most popular choice for implementation of large programming projects. However, these languages are insufficiently suitable for implementation of large software projects, in which one or several

problems often belong to the symbolic processing domain where non-traditional languages, such as Lisp or Prolog, are more adequate. For instance, development of a database with a complex structure (approx. a hundred of various relations) and with a natural language interface (queries written as sentences from a restricted subset of natural language and responses in a convenient NL form) involves the following problems to be resolved:

Problems	***Suitable programming languages***
Syntactic analysis of NL query	Refal
Semantic analysis of the query	Lisp, Prolog
Processing of the query	Prolog
Elaboration of response	Lisp, Refal
Modern user interface	C++, Object Pascal, Java
DB managing operations	C++

On the right hand of the problems, corresponding adequate programming languages are indicated. Evidently, languages oriented to symbolic processing are preferable for syntactic and semantic analysis of natural language queries, as well as for generation natural language phrases expressing responses. We suppose that semantic analysis of the queries may imply some logic inference, which is available in Prolog.

Thus, in order to facilitate implementation of programming projects an integration of programming techniques from different languages and styles is required. In particular, it seems attractive to enhance the power of popular imperative object-oriented languages with special data structures and control mechanisms from non-traditional languages.

As far as the necessity of integration of various programming techniques arisen long before the appearance of popular object-oriented languages, two simplest ways were proposed for solving the problem. The first way suggests creating in the source language some procedural analogue of the necessary technique from another language. This way is labor-intensive and does not preserve the primary syntax and semantics of built-in techniques.

Another way of integration involves coding each problem in an appropriate programming language and integrating of resulting program modules with the aid of multitask operating system (for example, via initiating its own process for each module). This way is difficult to realize because of closed nature of implementation of non-traditional languages and incompatibility of data structures from different languages (e.g., data structures of Prolog and C++ are incompatible).

However, the first way of integration was recently developed for integrating various programming techniques on the basis of an imperative object-oriented language, such as C++ or Object Pascal. The key idea of the method proposed in [Bolshakova and Stolyarov, 2000] is to design, within the given object-oriented language, special classes of objects and operations modeling necessary data and control structures from another language. The method was successfully applied for building functional techniques of the programming language Lisp [Steele, 1990] into the C++ language [Stroustrup, 1997], resulting in a special library of C++ classes that permits to write Lisp-like code within a C++ program.

We should also note that the second way of integration, i.e. direct integration of programming codes written in different languages, now becomes perspective in connection with the development of Microsoft.NET platform, which permits compiling and linking such different codes.

Learning Programming Paradigms

Necessity of integrating various programming techniques and languages within the same software project is the claim of modern programming. Therefore, a profound education in the field of computer science should be based on learning programming techniques of different paradigms. This implies learning several different algorithmic

languages, as none of languages can comprise all possible techniques from various programming styles. Our point is that courses on modern programming languages included in typical curricula should be complemented by special lectures devoted to programming paradigms and intended to compare their base programming techniques and to explicate distinguishing features of the paradigms. Another option to deepen the knowledge of programming techniques and programming languages is to enrich a general course on programming languages with education material on the main programming paradigms. Since the 80s a similar course is read at Algorithmic Languages Department of Faculty of Computational Mathematics and Cybernetic in Moscow State Lomonossov' University.

The importance of learning programming paradigms is also explained by the fact that we cannot know the future of popular modern languages. During the history of programming, many languages became dead, while some other languages have lost their popularity, so some modern languages may have the same destiny. However, the main programming paradigms will be the same, as well as their base programming techniques, and thus their learning is a permanent constituent of education in the field of computer science.

Conclusion

We have outlined main programming paradigms, as well as programming techniques and programming languages elaborated within them. Programming techniques of traditional imperative paradigm essentially differ from techniques of nontraditional ones – functional and logic. They have different scopes of applicability, and for this reason necessity to integrate techniques of different paradigms often arises in programming projects. Accordingly, a profound education in computer science implies acquirement of programming techniques of all main paradigms, and usual learning of modern programming languages should be complemented by learning of programming paradigms and their base programming techniques.

Bibliography

[Bolshakova, Stolyarov, 2000] Bolshakova, E., Stolyarov A. Building Functional Techniques into an Object-oriented System. In: Knowledge-Based Software Engineering. T. Hruska and M. Hashimoto (Eds.) Frontiers in Artificial Intelligence and Applications, Vol. 62, IOS Press, 2000, p. 101-106.

[Clocksin, 1984] Clocksin, W.F., Mellish C.S. Programming in Prolog, 2nd edition. Springer-Verlag, 1984.

[Goldberg, 1982] Goldberg, A., Robson D. Smalltalk-80 – the language and it's implementation. Addison-Wesley, 1982.

[Field, 1988] Field, A., Harrison P. Functional Programming. Addison-Wesley, 1988.

[Finkel, 1996] Finkel, R.A. Advanced Programming Language Design. Addison-Wesley Publ. Comp., 1996.

[Steele, 1990] Steele, G. L. Common Lisp – the Language, 2nd edit. Digital Press, 1990.

[Stroustrup, 1997] Stroustrup, B. The C++ Programming Language, 3rd edition. Addison-Wesley. 1997.

Authors' Information

Elena I. Bolshakova – Moscow State Lomonossov University, Faculty of Computational Mathematics and Cybernetic, Algorithmic Language Department; Leninskie Gory, Moscow State University, VMK, Moscow 119899, Russia; e-mail: bolsh@cs.msu.su

EDUCATIONAL MODEL OF COMPUTER AS A BASE FOR INFORMATICS LEARNING

Evgeny Eremin

***Abstract**: It is proposed to use one common model of computer for teaching different parts of the informatics course, connected with both hardware and software subjects. Reasoning of such slant is presented; the most suitable themes of the course, where it is practical, are enumerated. The own author's development (including software support) – the educational model of virtual computer "E97" and compiler from Pascal language for it – are described. It is accented, that the discussed ideas are helpful for any other similar model.*

***Author Keywords**: education, teaching, learning, informatics, computer model, compiler.*

***ACM Classification Keywords**: K.3.2 Computers and education: computer and information science education; I.6.3 Simulation and modeling: applications*

Introduction

One of essential and principled difficulties in studying informatics course is the complex character of the contents of this young and fast-developing science. In particular, a computer itself, being in point of fact the indissoluble unity of various technologies, for simplicity of study is divided into software and hardware (see the review of computer subjects in [1]). The distance between these two groups of disciplines shows a tendency to grow: numerous program layers (ROM BIOS, operating system, high-level languages and visual systems, application software) increasingly separate people, who work on computer, from hardware. Kaleidoscopic interchange of hardware models and software versions complicates the selection of material to study yet more.

As a result, in the conventional courses we see user interface and elements of the high-level language on the one hand, and computer machinery with its binary, absolutely "invisible" for user in practice, on the other. Qualified lecturer certainly demonstrates and accents their linkage, but not all students thoroughly recognize this actual unity. Unfortunately, the mentioned above separation increases every year, and it makes more difficult to form adequate students' outlook.

One of the possible ways of logical join of all courses, connected with software and hardware, is to study them on some common base. As real computers has complex, different in details and fast-changing organization, the deepest courses often replace computer hardware with a simpler educational model [2-5]. Such models are demonstrative and easy to understand on the one hand, and retain all most important and invariable features of real machines on the other one. Apparently the most particular model, called MIX, was created by the famous mathematic Donald Knuth as some abstract base for studying the fundamental principles of programming and computer calculations [2]. It's worth attention that recently the author of this classical multivolume book has upgraded his model [6, 7] to make it much more actual.

The aim of this paper is to demonstrate that the educational model of computer may be used in informatics wider and practically become a connecting-link between learning of software and hardware. Although the author used his own educational model "E97" [8, 9] first, all developed ideas can be fully realized on any other similar model.

Possible Usage of Educational Models in Informatics Learning

The most evident part of the course, where teaching may be based on an educational model, is computer organization. As every well-constructed model certainly reflects the most essential features of its original, this subject needs no proof.

Another suitable theme is number notations, where we can explore the theoretical algorithms of converting from, for example, decimal system into binary or vice versa, on our model base. The technique of model's usage depend on pedagogical aims: students may develop the program for converting themselves, analyze a code

which was prepared for them, test a ready program especially for the most interesting cases (signed or unsigned data, negative numbers, large values and overflow) and so on. Consequently we obtain a possibility to learn this matter in practice instead of dull abstract exercises on the paper sheet.

Expanding these ideas, we may offer to learn non-numeric information coding (text, graphics) with the help of an educational computer model too. Several other fundamental problems – byte organization of memory, storing of multi-byte values (big or little endian mode), ASCII or Unicode character sets – will necessarily be discussed and practically assayed on the side.

The detailed study of logical operations such as AND, OR, XOR, NOT on the base of the model is also an interesting application. This is particularly important because except high-level logical operations with Boolean data (widely used in all conditions and queries), computer instruction set always contains bitwise commands with the same names.

At last the same educational computer model can serve as a base of software study. Foremost we can mention fundamental ideas of compiling (will be demonstrated below), text proceeding, data compressing and other software foundations. The visual analysis of several samples on the computer model (program of changing letters' case, converting number to string and vice versa, compressing some sequences of identical codes, advantage of variable-length coding) often improve students' knowing much better than several abstract lectures or solving a lot of problems with the help of application software.

If an educational model of computer is developed enough, we may even try to demonstrate logic of the parallel computations or multitasking.

So the main novelty in the described slant is not the idea to use computer models in hardware learning (they are traditionally applied in this field), but the possibility to study other subjects of the informatics course on the common base of one educational model. In particular such method of teaching for software subjects is not practiced yet.

"E97" as an Example of the Base Model

It was already accented above that the concrete choice of the model for learning is not too essential. Nevertheless an educational model must closely accord to the contents of informatics course, be simple and demonstrative, but on the same time contain the most representative features of the modern computers. Hence not every model of computer is suitable as a common base of the course.

Making no pretence to the unicity of solution, in 1997 the author developed an educational model called "E97" [8]. Its description was later published in the books with greater edition [4, 9]. The model is used in the teaching of different themes of the informatics course in several educational institutions in Russia.

The following features, inset into "E97", differentiate it from other ones:

- adequacy to real principles of organization of personal computers (byte memory structure, information exchange via input/output ports and so on);
- very simple, but full-range instruction set;
- possibility to process non-numeric data with different memory dimension;
- actual memory addressing, including indirect method by means of processor's registers;
- wide use of the subroutine library that simulate ROM-BIOS; these subroutines can additionally demonstrate the samples of programming;
- several levels of learning from introductory to full-scale processor language programming.

The architecture of the virtual computer "E97" is similar to the legendary PDP computer family [10], known by its exact consistency and clearness. It consists from processor unit, two kinds of memory – RAM and ROM, and also simulates hardware exchange with keyboard and display. "E97" model has a capability to process numeric information (two bytes) as well as text (one byte) data.

Thoroughly developed ROM is another important feature of the model. Its existence essentially facilitates student's work with external devices, actually reducing it to the standard subroutine call. Such technique really takes place in modern computers: in IBM PC, for example, this kind of memory is called ROM-BIOS [11].

From the educational point of view it's important to accent, that ROM contents is stored in the text file with detailed comments, so such presentation may be used as a learning subject.

Being a multilevel model, "E97" allows creating the individual tasks of different complexity; hence this educational model may be useful in different kinds of educational institutions.

"E97" has several software realizations, including MS-DOS and Windows versions. The most universal realization was written on JAVA language, so it doesn't depend on hardware and able to run on any computer. The last mentioned version possesses one more advantage: its JAVA applet can be inserted into any Web-page with educational materials.

Demonstrational Compiler, Based on the Educational Model

The educational model of computer can be used as a base for the familiarity with the software principles. The importance of such slant springs out in many respects from the way of familization with computers at present. The question is that the computer specialists with long experience perfected their knowledge parallel to extension of calculating machinery. So they have naturally passed the way from processor codes to modern high-level languages and have seen full pallet of programming methods. Now most of people begin to study straight from high-level languages (or even fully ignore them, so the computer logic becomes mysterious for them). As a result, this missing of several technologies makes many concepts, such as methods of passing parameters or constructing of economical data structures, unappreciated.

To compensate the described above limitations, the author offers his specialized educational software, which demonstrates the major principles of automatic program generation. The professional systems are evidently unsuitable for mentioned above educational purposes, because they are absolutely closed and generate the code that is hard for human analyses.

The educational demonstrational compiler "ComPas" operates with some limited subset of Pascal language. This subset includes all algorithmical structures: assignment, conditional operator IF and three traditional types of cycles – WHILE, REPEAT and FOR. Standard input/output procedures READ and WRITE are realized as the call of subroutines from the special library; these subroutines in turn make the necessary preparations and redirect the call to ROM. The compiler supports standard data types and arrays of them.

The enumerated above possibilities allow to demonstrate our students the following essential features of high-level languages:

- variables, constants, typed constants and difference between them;
- different data types and organization of their storage in RAM (including arrays and access to their elements);
- conversion of values from one type into another (CHAR into INTEGER or so on);
- methods of the main algorithmical structures' realization;
- details of procedures usage

and other fundamental concepts and principles.

To illustrate how demonstration compiler works, let's consider the simplest Pascal program, presented below:

```
PROGRAM sample;
CONST x = 2;
VAR y: INTEGER;
BEGIN y := x + 10;
      WRITELN(y)
END.
```

As a result of translation, "ComPas" generates short and transparent code and shows it on the screen with detailed comments in the form of a table. The example of such table, shown below, is filled for Intel processor's codes (although the program for educational model looks easier, nevertheless its description requires more additional information, so we'll not discuss it in this paper).

Address	Code	Assembler	Actions	Comments
100	E97D01	jmp 0280		jump to the beginning
103	…			library with standard subroutines
280	B80200	mov ax,0002	2 ==> ax	constant ***x***
283	B90A00	mov cx,000A	10 ==> cx	constant 10
286	01C8	add ax,cx	ax + cx ==> ax	***x*** + 10
288	A3FE04	mov [04FE],ax	ax ==> [4FE]	save result into ***y***
28B	A1FE04	mov ax,[04FE]	[4FE] ==> ax	load ***y*** value
28E	E8C6FE	call 0157	print integer	WRITE ***y*** (call subroutine from the library)
291	E8FDFE	call 0191	next line	LN (call subroutine from the library)
294	CD20	INT 20	return to system	END.

The analysis of the above program comes easy and cogitable even for beginners. Students can simply find every Pascal operator and carefully examine it (in the above table separate operators are marked by gray color; software has special navigation controls for this purpose).

The educational compiler is freely spread via the Internet and everybody can download it from Web-page [12]. You also may find links for detailed on-line documentation there.

Conclusion

Thereby we see that the idea of the usage of common educational model of computer as a base for learning is suitable for different themes of the informatics course. The discussed slant, as pedagogical experience attests, gives a possibility to refine students' knowledge and forms in their mind a more adequate picture of data processing by means of modern computer machinery.

Bibliography

[1] Computing Curricula 2004 (draft). URL: http://www.acm.org/education/curricula.html

[2] Donald E. Knuth. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley, 1997

[3] J. Glenn Brookshear. Computer Science: an overview. Reading, Massachusetts: Addison-Wesley, 2000

[4] Mogilev A.V., Pak N.I., Henner E.K. Informatics. Moscow: Academy, 1999 (in Russian)

[5] Lapchik M.P., Semakin I.G., Henner E.K. Methodics of informatics teaching. Moscow: Academy, 2001 (in Russian)

[6] Donald E. Knuth. MMIXware: a RISC Computer for the Third Millennium. Heidelberg, Springer-Verlag, 1999

[7] MMIX Homepage. URL: http://www-cs-faculty.stanford.edu/~knuth/mmix.html

[8] Eremin E.A. How modern computer works. Perm: PRIPIT, 1997 (in Russian)

[9] Eremin E.A. Popular lectures about computer organization. St.-Petersburg: BHV-Petersburg, 2003 (in Russian)

[10] Wen C. Lin. Computer Organization and Assembly Language Programming: PDP-11 and Vax-11. New York: John Wiley & Sons, Inc., 1985

[11] Peter Norton. The Peter Norton Programmer's Guide to the IBM PC. Microsoft Press, 1985

[12] ComPas Hompage. URL: http://www.pspu.ru/personal/eremin/eng/myzdsoft/compas.html

Author's Information

Evgeny Aleksandrovich Eremin – Perm State Pedagogical University. Russia, 614990, Perm, Sibirskaya str., 24. e-mail: eremin@pspu.ac.ru

ADAPTIVE ROUTING AND MULTI-AGENT CONTROL FOR INFORMATION FLOWS IN IP-NETWORKS

Adil Timofeev

Abstract: *The principles of adaptive routing and multi-agent control for information flows in IP-networks.*

Keywords*: telecommunication system, adaptive quality service, multi-agent control, IP-network.*

ACM Classification Keywords*: I.2.11 Distributed Artificial Intelligence: Multiagent systems; F.1.1 Models of Computation: Neural networks*

Introduction

Evolution of informational and telecommunication system requires on the modern stage a development of theoretical bases of design for integrated infotelecommunication computer networks of new generation, including telecommunication systems (TCS) and distributed information and computer resources (local and regional computer networks, data stores, GRID-systems etc.). At this global TCS play role of tools for providing for users as external agents (clients) quality of service (QoS) for their plural access to information and computing resources, distributed in all over the world.

Improvement of global TCS is connected firstly with further development of theoretical bases and realization of methods of automation, optimization and intellectualization of systems for network control of information flow. Reason of it is that today network control network of global TCS depends significantly on network administrators and operators. However, their possibilities and abilities are limited principally.

Alternative way for improvement of network flows control in global TCS is its automation on the base of dynamical models of TCS as complex network plants of control, methods of optimization of processes of routing of data flows and principles of adaptive and intelligent control for traffic with use of multi-agent technologies and protocols of new generation (for example, IPv6-protocols). On this new way there is a possibility of consideration either only real dynamics of TCS, i.e. real change of structure (topology) and parameters (weights of communication channels) of TCS in a real time or adaptation to different factors of uncertainty on the base of monitoring and functional diagnosis of TCS.

In the paper dynamical models of global TCS with changing structure, mathematical models, optimization algorithms and protocols of dynamical, adaptive, neural and multi-agent routing of data flows are described. These models, methods and protocols of new generation are important part of modern theory of adaptive and intelligent control for information flows in global TCS. They reflects experience and scientific reserve, obtained in a process of fulfillment in 2002-2004 a state contract №37.029.0027 with the title “Adaptive methods for control of data flows in telecommunication systems of new generation” of Russian Federal Agency on Science and Innovations in the framework of Federal Scientific Technical Program “Researches and development on priority directions for science and technical development”.

1. Evolution of TCS and Intellectualization of Network Control

Globalization and other modern tendencies of TCS development cause not only significant overview of basic telecommunication concepts but also significant technological drifts as follows [1-3]:

- from speech traffic to data traffic and multimedia traffic;
- from special TCS to global TCS of new generation;

- from local special service to multimedia universal service and applications with guaranteed quality in every time and everywhere.

With consideration of existing tendencies, two scenarios of TCS development are possible: revolutionary and evolutionary. Revolutionary scenario is in a fast substitution of existing electronic TCS development by optical systems with channel capacity about thousands Gb/s. Realization for this scenario requires very big investigations for development and mass production of standardized optical components for global TCS. Evolutionary scenario for TCS development is based on their gradual modernization by advancement of systems, protocols and technology of control for transferred information flows. Today realization of this scenario goes very fast and causes creation of corporative and global TCS of new generations.

Fast grow of real time traffic and its heterogeneous (multimedia) character cause network collisions and overloading, blocking normal TCS work. Rapid development of new kinds of service (electronic commercial, electronic games and entertainment etc.) increased sharply requirements for quality of service (QoS) and information security.

Appeared problems caused necessity for creation of new mathematical models, algorithms of routing and protocols, providing solution of the following tasks:

- development of scaled address system;
- optimization for processes or data flows routing;
- providing of guaranteed quality of service (QoS);
- support and realization of mobile service and Internet.

New tasks and requirements for IP-technologies caused wide use of fourth version of classical protocol (IPv4) and development of its sixth version (IPv6), providing the following [1-2]:

- increase of address field of working part of package till 128 bits, that increases quantity of IP-addresses till 1020 on every TCS node;
- increase of length of package title till 320 bits with information localization, necessary for router work;
- increase of effectiveness by aggregation of addresses and fragmentation of big packages;
- providing of security of information by authentication of nodes-sources and nodes-receivers of information, coding and keeping of wholeness of transferred data.

Architecture of global TCS consists of four basic (basis) subsystems [3]:

1. Distributed communication system (DCS);
2. Network control system (NCS);
3. Distributed information system (DIS);
4. Distributed transport system (DTS).

These subsystems are connected each other and destined for controlled transfer to users of global TCS distributed and computing resources, stored in CS [7-10].

NCS of global TCS of new generation should be adaptive and intelligent [3-11], i.e. have abilities to:

- adaptation (automated self-adjustment) with relation to changing quantity of users, their queries and requirements to quality of provided service, changing structure (topology) of TCS and parameters of nodes and communication channels etc.;
- learning and self-learning for new functions and rules for TCS work;
- self-organization of structure and functions of NCS with dependence on TCS changes;
- prediction for network collisions and other sequences of network control.

Thus, adaptivity and intelligence are the most important features of perspective NCS, destined for regional and global TCS of new generation.

2. Optimization of Traffic Control and Routing of Information Flows

Problem of traffic control in global TCS is decomposed on two interconnected tasks:

1) Planning, optimization and adaptation of routes for flows transfer between TCS nodes;
2) Control of transfer for data flows on defined route with adaptation to changing traffic, possible overloading or changes of TCS topology.

Simplest static setting of task for planning and optimization of data transfer routes is based on suggestion that TCS structure (number of nodes, topology and communication channel cost) is known and constant and TCS internal agent –user is played by one client, forming query to one of node network computers.

Dynamical setting of routing task by user query considers that TCS structure may change with current time but reminds known. At this network information automatically renewals in definite time intervals (Time to Live, TTL), that causes corresponding change of optimal routes.

At adaptive setting of task routing is made in uncertainty conditions, when topology of communication channels and TCS traffic may change unpredictably and accessible information has local character, monitoring and renewal of network information allows to correct routes, adapting them to changing conditions of TCS work (network overloading, failures etc.).

Necessity in dynamical and adaptive routing of data flows in global TCS appears in the following cases:

1) change of cost of TCS communication channel (for example, at their substitution);
2) failure of one or several communication channels in TCS;
3) addition to TCS new communication channels;
4) failure f one or several communication nodes of TCS;
5) addition to TCS new nodes;
6) overloading of TCS communication channels;
7) overloading of buffers of TCS nodes.

In the first case, it is necessary to renewal data about weights (costs) of TCS graph edges, and in the second and third ones – to eliminate or add corresponding edges in TCS graph. In the fourth and fifth cases, it is necessary to change data about TCS nodes by elimination or addition of corresponding nodes in TCS graph. In the sixth and seventh cases corresponding edges and nodes of TCS graph are shown as “prohibited” communication channels and nodes, playing role of unavoidable obstacles for routing and data flows transfer.

Multi-agent setting of task requires development for methods of static, dynamical and adaptive routing in conditions of multi-address transfer and collective use of TCS, when number of external agents-clients of TCS and quantity of their queries may change unpredictably with current time. In this case network overloading and collisions may appear, for their avoidance or resolution special algorithms and tools for their realization are necessary [4-6].

Suggested methods of solution for network control are based on development of adaptive, neural and multi-agent routers, using protocols of new generation (for example, IPv6-protocols).

3. Models and Methods of Adaptive and Multi-agent Routing

Necessity in adaptive routing of data flows appears at unpredictable structure changes (topology of nodes and communication channels) or parameters of global TCS and also at overloading of node buffers or TCS communication channels. Thus routers should plan and correct optimal routes of data package transfer, adapting them to possible TCS changes, appearing in real time.

For that, it is necessary to have feedback communication about current state of nodes and TCS communication channels, which may be organized with the help of monitoring and information exchange between TCS nodes.

Adaptive routing of data flows in global TCS has a series of advantages in relation with non-adaptive (static or dynamic) routing, as follows:

- provides workability and reliability of NCS at unpredictable changes of their structure or TCS parameters;
- causes to more uniform loading of nodes and TCS communication channels by “equalizing” of load;
- simplifies control for data flow transfer at network overloading;
- increases time of dependable work and TCS productivity at high level of provided service at unpredictable changes of network parameters and TCS structure.

Achievement of these advantages depends significantly on used principles, models and algorithms of adaptive routing of TCS data flows with unpredictably changing structure and beforehand unknown traffic.

Principle of adaptive routing with local feedback communication from one node is that data package is transferred into communication channel with the shortest queue or with the most probability of channel preference. At that local equalizing of load in output TCS channels may take place. However, in this case deviation from optimal route is possible. More effective principles of adaptive routing are based on transfer of local information (feedback communication) from neighbour nodes or global information from all TCS nodes. Such information may be presented by data about failures or delays in nodes or communication channels in TCS.

Models and principles of adaptive routing of data flows in global TCS may be divided on three classes [3,5-7]:

- centralized (hierarchical) routing;
- decentralized (distributed) routing;
- multi-agent routing.

Principle of centralized routing is that at first every TCS node transfers information about its state (delay or output channel capacity) to central node-router. Then this router calculates optimal route on the base of obtained global information about current TCS state and transfer it back to all route nodes. After that controlled data package transfer from node-source to node-receiver on planned optimal route begins.

Principle of decentralized routing is based on change of local information between TCS nodes and use of this information about current state of nodes and TCS communication channels for calculation of local-optimal route. As subsequent plots of this route are calculated distributed controlled package transfer from node-source to node-receiver of TCS is realized.

Conclusion

Suggested in the paper mathematical models and optimization methods of dynamical, adaptive, neural and multi-agent (multi-address and multi-flow) routing of information flows for global TCS of new generation are important step towards creation of the theory of adaptive multi-agent (mass) service of global informational and telecommunication networks, which should change traditional theory of mass service. They may be useful for organization of adaptive multi-agent (mass) service of GRID-infrastructure of different scale and destination or for creation of new generation of scientific-educational networks.

Work has been done at partial support of the grant of RFBR № 03-01-00224a, the grant of RHSF № 03-06-12016b and the RAS Presidium Program “GRID”.

Bibliography

[1] Heleby S., Mac-Pherson D. The Routing Principles in Internet. 2-nd Edition: Translation from English. - Moscow, Publishing House “Williams”, Urbana, 2003. – 448p.

[2] Stallings V. Modern Computer Networks. – Saint-Petersburg: Piter, 2003, - 783 p.

[3] Timofeev A.V. Problems and Methods of Adaptive Control for Data Flows in Telecommunication Systems. – Informatization and Communication,№ 1, 2003,pp. 68-73.

[4] Timofeev A.V. Methods of High-Quality Control, Intellectualization and Functional Diagnostics of Automated Systems. – Mechatronics, Automation, Control, 2003, № 2, pp. 13-17.

[5] Syrtzev A.V., Timofeev A.V. Neural and Multi-Agent Routing in Telecommunicational Networks. – International Journal “Information Theories and Their Applications”, 2003 , № 2, pp 167-172.

[6] Timofeev A.V. Models for Multi-Agent Dialogue and Informational Control in Global Telecommunicational Networks. – International Journal “Information Theories and Their Applications”, 2003, № 1, pp. 54-60.

[7] Timofeev A.V. Architecture and Principles of Design of Multi-Agent Telecommunication Systems of New Generation. – Proceedings of 11-th All-Russian Scientific Methodical Conference “Telematika-2004” (Saint-Petersburg, 7-10 June 2004), vol. 1, pp. 172-174.

[8] Timofeev A.V., Ostyuchenko I.V. Multi-Agent Quality Control in Telecommunication Networks. – Proceedings of 11-th All-Russian Scientific Methodical Conference “Telematika-2004” (Saint-Petersburg, 7-10 June 2004), vol. 1, pp. 177-179.

[9] Timofeev A.V. Multi-Agent Information Processing and Adaptive Control in Global Telecommunication and Computer Networks. – International Journal “Information Theories and Their Applications”, 2003, № 10, pp. 54–60.

[10] Timofeev A.V. Intellectualization for Man-Machine Interface and Network Control in Multi-Agent Infotelecommunication Systems of New Generation. – Proceedings of 9-th International Conference “Speech and Computer”(20–22 September, 2004), Saint-Petersburg, Russia, pp. 694–700.

[11] Timofeev A.V., Syrtsev A.V., Kolotaev A.V. Network Analysis, Adaptive Control and Imitation Simulation for Multi-Agent Telecommunication Systems. – Proceedings of IFAC International Conference on Physics and Control 2005 (August 24-26, 2005, Saint-Petersburg, Russia).

[12] Timofeev Adil. Adaptive Control and Multi-Agent Interface for Infotelecommunication Systems of New Generation. – International Journal “Information Theories and Their Applications”, 2004, № 11, pp.329–336.

Author's Information

Adil Timofeev – Saint-Petersburg Institute for Informatics and Automation; P.O.Box: 199178, 39, 14-th Line; Saint-Petersburg, Russia, e-mail: tav@iias.spb.su

TABLE OF CONTENTS OF VOLUME 12, NUMBER 3