



**I T H E A**



**International Journal**

**INFORMATION THEORIES  
&  
APPLICATIONS**



**2005 Volume 12 Number 4**



**International Journal**  
**INFORMATION THEORIES & APPLICATIONS**  
 ISSN 1310-0513  
 Volume 12 / 2005, Number 4

Editor in chief: Krassimir Markov (Bulgaria)

**International Editorial Staff**

Chairman: Victor Gladun (Ukraine)

Adil Timofeev	(Russia)	Larissa Zainutdinova	(Russia)
Alexander Eremeev	(Russia)	Levon Aslanian	(Armenia)
Alexander Kleshchev	(Russia)	Luis F. de Mingo	(Spain)
Alexander Kuzemin	(Ukraine)	Martin P. Mintchev	(Canada)
Alexander Palagin	(Ukraine)	Milena Dobрева	(Bulgaria)
Alexey Voloshin	(Ukraine)	Laura Ciocoiu	(Romania)
Alfredo Milani	(Italy)	Natalia Ivanova	(Russia)
Anatoliy Shevchenko	(Ukraine)	Neonila Vashchenko	(Ukraine)
Arkadij Zakrevskij	(Belarus)	Nikolay Zagorujko	(Russia)
Avram Eskenazi	(Bulgaria)	Petar Barnev	(Bulgaria)
Boicho Kokinov	(Bulgaria)	Peter Stanchev	(Bulgaria)
Constantine Gaidric	(Moldavia)	Plamen Mateev	(Bulgaria)
Eugenia Velikova-Bandova	(Bulgaria)	Radoslav Pavlov	(Bulgaria)
Frank Brown	(USA)	Rumyana Kirkova	(Bulgaria)
Galina Rybina	(Russia)	Stefan Dodunekov	(Bulgaria)
Georgi Gluhchev	(Bulgaria)	Tatyana Gavrilova	(Russia)
Iliа Mitov	(Bulgaria)	Valery Koval	(Ukraine)
Jan Vorachek	(Finland)	Vasil Sgurev	(Bulgaria)
Juan Castellanos	(Spain)	Vitaliy Lozovskiy	(Ukraine)
Koen Vanhoof	(Belgium)	Vladimir Jotsov	(Bulgaria)
Krassimira Ivanova	(Bulgaria)	Zinoviy Rabinovich	(Ukraine)

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](http://www.foibg.com/ijita).

The camera-ready copy of the paper should be received by e-mail: [foi@nlcv.net](mailto: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](http://www.foibg.com)).

International Journal "INFORMATION THEORIES & APPLICATIONS" Vol.12, Number 4, 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](http://www.foibg.com), e-mail: [foi@nlcv.net](mailto:foi@nlcv.net)

© "Information Theories and Applications" is a trademark of Krassimir Markov

Copyright © 1993-2005 FOI-COMMERCE, Publisher

Copyright © 2005 For all authors in the issue.

All rights reserved.

ISSN 1310-0513

---

## LOGICAL MODELS OF COMPOSITE DYNAMIC OBJECTS CONTROL

Vitaly Velichko, Victor Gladun, Gleb Gladun, Anastasiya Godunova,  
Yurii Ivaskiv, Elina Postol, Grigorii Jakemenko

*Abstract:* The questions of designing multicriteria control systems on the basis of logic models of composite dynamic objects are considered.

*Keywords:* control, logical model, composite dynamic object, balancing network.

*ACM Classification Keywords:* I.2.8 Problem Solving, Control Methods, and Search: Control theory; F.1.1 Models of Computation: Neural networks

---

### 1. Introduction

---

The special attention in the class of control systems is deserved by multicriteria control systems of composite dynamic objects (CDO). As composite dynamic objects we shall consider such technical objects with a broad spectrum of states and plenty of permissible transitions between states, the control by which one consists in implementation of often and miscellaneous control actions in connection with often changes modes of exploitation, purposes, conditions of operation, preferences of the user, estimation's criterions.

It is necessary to take into account the following conditions at designing control systems of such objects.

1. A basic function of control systems is the selection of control actions ensuring optimization of a state of control object. For implementation of this function in multicriteria systems, the methods of vectorial optimization are used. In the basis, which one lay selection operations of alternatives by confrontation of their vectorial estimations by given criterions. The run time of these base operations depends first of all on organization of structure presenting the information on alternatives and criterions of estimation in memory of a management system. The structures used for representation of information should provide at all stages of control procedure selection of indispensable alternatives or criterions of estimation without looking through appropriate sets. The factor of processing speed is specially important for control in extreme situations. The designing of information structures accelerating processes of selection, is a problem of logical designing [1-3].

2. The problems of multicriteria vectorial optimization have no precise solution more often. Usually process of the initial problem solution ends getting sequence of effective not predominant alternatives (Pareto set). Further "improvement" of the solution can be obtained only at the expense of calling the additional information from person who takes decision [Decision taking Person](PTD). Attempts to "improve" the solution are implemented by means of sort out alternatives or criterions reflecting preferences PTD. An effective means of selection the "best" solutions become testing of versions proposed PTD. PTD preferences should be envisaged in a structure of a control system at a phase of its logical designing. For preliminary modelling and selection of the control solutions, which are taking into account preference PTD, it is useful in design organizations to use a special modelling solutions stand.

3. Formation of the control solutions is a complex process including such operations, as monitoring and estimation of a situation, modelling, estimation and selection of control operations.

Logical model of control of objects has to:

- Allow effective execution of the basic procedures of control;
- Assist the acceleration of the search-sorting operations;
- Contain the information about actions, variants of actions, sequence of actions, connections of actions with objectives, i.e. as a matter of fact, to show all the totally procedural knowledge about objects of control;
- Represent the states of the object of control that arise on the separate steps of solution of the task of control;

- Allow the description to terminology of the ending user;
- Allow the enter of the changes under the influence of information that comes in from without.

In the time of developing means of automation of complex control procedures, it is desirable to find a unified effective technical principle, which one would provide enough fast response time all components of complex process. In concordance with methodology of an artificial intelligence, such determining principle can be idea of organization representation and looking up of knowledge.

In modern intellectual systems such successful idea are traditionally connected to usage of network ways of a data representation, that allows to avoid having overcome large amounts of information during execution of search operations.

Report deals with logical designing of multicriteria control systems of composite dynamic objects on the basis of balancing networks [4-6].

---

## 2. Problem of Control

---

It is comfortably to consider the processes of choice of handling actions as a task of satisfaction of limitations (CSP - Constraint Satisfaction Problem). In the tasks of this class, states are described by sets of values of variables, and purposes are a set of limitations, with which these values must satisfy.

Let's consider a problem of such type.

There is an object of control

$$\langle X, P, D, Y, F, L \rangle, \quad (1)$$

where:

$X$  - set of entrance variables - set of numerical, Boolean or linguistic variables the values of which are not determined with the help of some functions or rules through other variables. In set  $X$  the subset  $C$  of *controlling* variables by which change it is possible to influence states of object of control is allocated. The controlling action changing its value is connected to each variable of set  $C$ ;

$P$  - set of *parameters* - numerical, Boolean or linguistic variables which values are considered set and constant during the decision of one task;

$D$  - set of *derivative* variables - numerical, Boolean or the linguistic variables determined through other variables functionally or with the help of rules such as « if - that »;

$Y$  - set of *target* variables,  $Y \subseteq D$ ;

$F$  - set of functions and rules « if - that », of derivative variables determining dependence of derivative variables on other variables;

$L$  - set of restrictions of type  $a_y \leq y \leq b_y$ , where  $a_y, b_y$  - numerical constants,  $y \in Y$ .

The task will consist in a finding in space of controlling variables of a point in which all restrictions of set  $L$  are carried out.

The target variable refers to *normalized* if its value corresponds to the set restrictions. The target variables, which are not corresponding to the restrictions, are considered *not normalized*. Controlling variables, which change causes change of value of a target variable  $y \in Y$ , refer to *relevant* as this target variable.

If mapping  $X \rightarrow Y$  it has been set by some system of the equations, for a choice of the decision it would be possible to use methods of mathematical programming. In (1) mapping  $X \rightarrow Y$  directly it is not set. The decision should be received only on the basis of the semantic information describing the environment and the purposes. The tool of the decision of such problems is balancing networks.

---

## 3. Balancing Models

---

A *balancing network* [1-3] is the oriented graph  $\langle V, U \rangle$  (illustrated by Fig.1), in which:

$V$  - set of the vertices corresponding to variables of all types in model (1).

$U$  - set of arches. The vertex corresponding to a variable  $d \in D$ , incorporates coming arches to the vertices representing variables which are used for calculation  $d$ .

To the left parts rules correspond Boolean variables which accept value 1 at performance of conditions of applicability of rules; the expressions which are included in the right parts of rules, are calculated only at value "1" of this variable. At use of a balancing network for representation of model (1) vertices, presenting controlling and target variables refer to accordingly controlling and target vertices of the network. Vertices proper to the derivative variables are provided with pointers, indicative functions or rules, which are used for determination of values of these variables.

The proper limitations from the set  $L$  contact with target vertices.

Alongside with such structures as «trees of decisions», «and-or graphs», «networks of utilities», balancing networks are the effective instrument of decision-making process. In the systems using balancing model, great value has the block providing automatic construction of the balancing network under the information (1), and also operative reorganization of the network at changes of the environment.

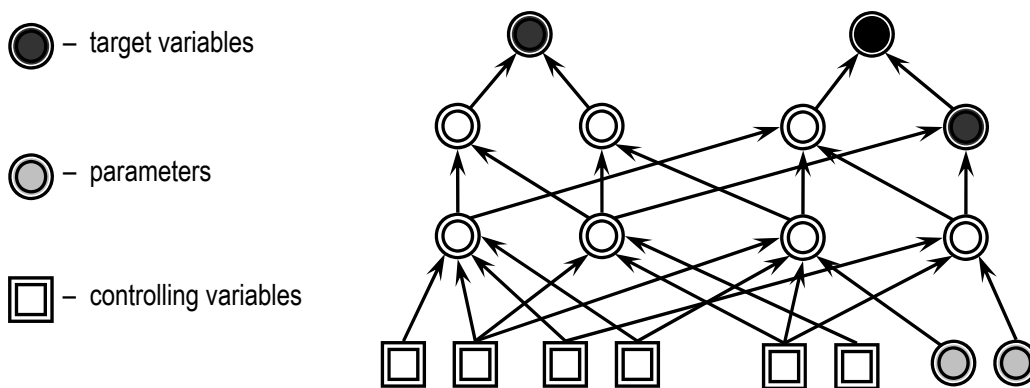


Fig. 1. A balancing network.

The balancing network is development of settlement - logic networks [4].

#### 4. Choice of Controlling Actions

Constant watching of values of entrance variables, and also calculation of values of derivative variables are carried out in the system of control. If the set of values of target variables corresponds to a known class of situations, the system specifies the recommended sequence of actions fixed in its memory. Each carried out action causes change of value of a corresponding controlling variable and, as consequence, change of values of all variables dependent on it. If the arisen extreme situation does not belong to any known class of situations, procedure of a choice of controlling actions as a matter of fact solves a problem of multicriterion decision-making.

For the description of procedure, we shall enter the following designations:

$Y'$  - set of not normalized target variables;

$R$  - set of controlling variables, relevant to not normalized target variables;

$h_r^y$  - a variable designating necessary "direction" of change by controlling variable  $r \in R$  for normalization of a target variable  $y$ ; the variable  $h_r^y$  - appropriates value 1 if value  $y$  is necessary to increase and value - 1 if value  $y$  needs to be reduced;

$H_r$  - the variable designating "direction" of change by controlling variable  $r$  in view of all not normalized target variables.

In procedure of a choice of controlling actions, the following operations are used.

Operation 1. Search of relevant controlling variables.

For each not normalized target, variable relevant controlling variables and the necessary direction of their change are defined by viewing against arrows of the ways coming into corresponding target top. As a result of performance of operation 1 set  $R$  is formed and values  $h_r^y$  are defined for all  $y \in Y'$ .

Operation 2. Calculation of values of variables  $H_r$ .

For everyone  $r \in R$  value of a variable  $H_r$  is calculated  $H_r = \sum_{y \in Y'} h_r^y$ .

If  $H_r > 0$ , it is expedient to increase value of a controlling variable  $r$ , if  $H_r < 0$  - to reduce.

Operation 3. A choice of the "best" controlling variable.

As the "best" the controlling variable having  $\max |H_r|$  can be chosen. This integrated criterion is used for a choice of the controlling action rendering normalizing influence on the greatest number of not normalized target variables. Applying various integrated criteria, it is possible to change strategy of control.

Operation 4. Modelling of application of the controlling action corresponding to the chosen controlling variable.

Controlling action changes value of the chosen controlling variable  $r$ . Depending on a sign  $H_r$  value  $r$  increases or decreases for some fixed size which is a priori underlined by experts for each controlling variable.

Values of all variables corresponding to vertices, which are connected by coming ways to the changed controlling variable, are recalculated.

Result of application of controlling action is normalization available and, probably, appearance of the new not normalized target variables.

For a new state of the network if in it still there are not normalized variables, all set above operations repeat.

In case of success procedure forms sequence of controlling actions, i.e. some plan of achievement of the normalized situation.

If changes of each of relevant controlling variables causes both positive, and negative changes of not normalized variables, it means, that full normalization of a situation with the help of the procedure is impossible, as, according to the theory of decisions, the space of not dominating alternatives (set of Pareto) is achieved.

Preferences of the person accepting the decision can be taken into account with the help of weights, which are entered for controlling or target variables and are used at a choice of controlling variables.

The described method is realized as program system.

Let's consider his application for the decision of the problems arising at operation of the helicopter.

The helicopter at absence of visibility as shows sad experience, is on occasion deduced on the modes leaving on corners and speeds of flight for established restrictions that result sometimes in destruction of the device and death of people. It is necessary on parameters of spatial position of the helicopter, modes of flight, works of the screw and engines to warn crew about danger and to give recommendations on controlling actions.

The specified task enters a class of problems where the system - on the basis of a balancing network can be used.

For decision of the task model (1) is formed.

1. Set of entrance variables:  $X$ .  $X_\theta, X_\beta, X_\gamma, \dots$  - controlling variables which, influencing on the screw, change parameters of the helicopter on corners  $\theta, \beta, \gamma$ , speeds  $V$ .
2. Parameters, which for separate modes of flight because of their slow change can be accepted as constants, among them are weight of the helicopter, height, external temperature, sometimes speed, etc.
3. Derivative variables - speed, corners, angular speeds -  $V, \theta, \beta, \gamma, \omega_x, \omega_{at}, \omega_z, \dots$  which can be determined through controlling variables and parameters. These variables are connected by the differential equations of the second order (the equation of movement of a firm body under action of external forces).
4. Restrictions on corners are set  $\theta, \beta, \gamma$ , to speeds, etc.

---

The decision of the task can be divided into two stages.

- Finding of a point in space of derivative variables in which all restrictions on corners and angular speeds are carried out.
- On the basis of limiting restrictions of dynamics of changes on corners and speeds, taking into account functional dependence of corners and speeds on moving controls the area of change of controlling variables is defined.

---

## 5. Summary

---

In the resulted procedure of formation multicriterion controlling decisions:

- search operations as much as possible become simpler due to the advanced associative opportunities of a balancing network;
- selection of controlling actions is carried out by the analysis of a target situation without taking into account any aprioristic information about utilities, priorities or probabilities of the purposes, conditions and actions though such information can be taken into account by giving weights to vertices and connections of a balancing network;
- the formed plan of action is directed on achievement of balance between conflicting criteria of a choice of the decision;
- the uniform organization of memory for a data storage and knowledge as a balancing network plays a role of the link connecting components of complex process in system and, thus, promoting achievement of compactness hardware and the software;
- processes of formation of a balancing network suppose an opportunity of operative reorganization of a network at a choice of decisions in dynamic environments.

On the basis of the described approach it is expedient to create the stands for the modelling and selecting controlling decisions in the projecting organizations.

---

## Bibliography

---

1. Zakrevskiy A.D., The Parallel Algorithm of the Logical Management, The Institute of the Logical Management, The Institute of the technical cybernetics NAN of Belarus, Minsk, 1999.
2. Alexeyev A.V., Borisov A.N., Sliadz N.N., Phomin S.A., Intellectual Systems of Decision making, "Zikatne", Riga, 1997.
3. Valkman U.R., Intellectual Technologies of The Searching Projecting, Port-Royal, Kiev, 1998.
4. Gladun V.P., The Partnership with a Computer. Human-machine Purposeful Systems. Kiev: «Port-Royal», 2000.
5. Gladun V., Vaschenko N., Control on the Basis of Network Models // Artificial Intelligence in Real-Time Control: 7th IFAC Symposium. - Arizona, USA. 1998.
6. Gladun V.P., Vashchenko N.D., Goal Formation and Action Planning on the Basis of the Network Model // Proceedings of the 1st Intern. IEEE Symp. "Intelligent Systems". Varna, Bulgaria, Sept.10-12. 2002, v.1, p.150-153.
7. Tyugue E. H., Conceptual programming. - M.: Nauka, 1984.

---

## Authors' Information

---

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

Gleb Gladun – Kamov Company, the 8th March 8a, 140007, Lubertsy, Moscow Region, Russia

Anastasiya Godunova, Yurii Ivaskiv, Elina Postol, Grigorii Jakemenko – National Aviation University, Prospekt Kosmonavta Komarova 1, 03058, Kiev, Ukraine

## THE ON-BOARD OPERATIVE ADVISORY EXPERT SYSTEMS FOR ANTHROPOCENTRIC OBJECT

Boris Fedunov

*Abstract:* A class of intelligent systems located on anthropocentric objects that provide a crew with recommendations on the anthropocentric object's rational behavior in typical situations of operation is considered. We refer to this class of intelligent systems as onboard real-time advisory expert systems. Here, we present a formal model of the object domain, procedures for obtaining knowledge about the object domain, and a semantic structure of basic functional units of the onboard real-time advisory expert systems of typical situations. The stages of the development and improvement of knowledge bases for onboard real-time advisory expert systems of typical situations that are important in practice are considered.

*Keywords:* expert systems, AI architectures, inference technique.

*ACM Classification Keywords:* H.4.2 Decision support.

---

### Introduction

---

In up-to-date complex engineering of man-machine objects, the problems of setting the goals of object operation (general and current) are always solved by a crew (an operator or a group of operators governing the man-machine object). Such objects are called anthropocentric. An anthropocentric object is a functionally consistent set of onboard measuring and executive devices (OM&ED), an onboard digital computing system (ODCS) with implemented algorithms for the anthropocentric object system-forming kernel (onboard digital computer (ODC) algorithms), and a crew with algorithms of its activity (CAA) supported by the information-control field (ICF) of a crew compartment, which is the carrier of everything listed above.

The anthropocentric object operation is naturally divided into operation sessions for which the anthropocentric object and its crew is prepared in advance. As a result of this preparation, prior information and the main (general) goal of the forthcoming operation session are transmitted aboard the anthropocentric object (inserted into the ODCS algorithms and into the crew activity algorithms).

---

### 1. Onboard Real-Time Advisory Expert Systems of Typical Situations for an Anthropocentric Object

---

The external world in which a TS ORTAES operates is the information medium of an anthropocentric object (Fig. 1) formed by the output information of onboard measuring devices, the standard (not included in the TS ORTAES) onboard digital computer algorithms, and the crew compartment information-control field signals. Before starting a forthcoming operation session [Fedunov,1998], *a priori* information about this session is loaded into the TS ORTAES. This information is transmitted from the intellectual system, preparing the anthropocentric object for the operation session [Vasil'ev and others, 2000].

For every problem sub-situation significant for a TS, the ORTAES generates solution recommendations for the crew with brief explanatory notes. The recommendations and explanatory notes for them come to the crew compartment information-control field (the information part). The crew can disregard the ORTAES recommendation and resolve the problem sub-situation by other methods without reporting about this to the ORTAES. In this case, the ORTAES must develop the next recommendation taking into account the method implemented by the crew. Any disregard by the crew toward ORTAES recommendations is registered by the onboard objective control system, and when the operation session is completed, this information is submitted to the intelligent system analyzing the completion of the operation session.



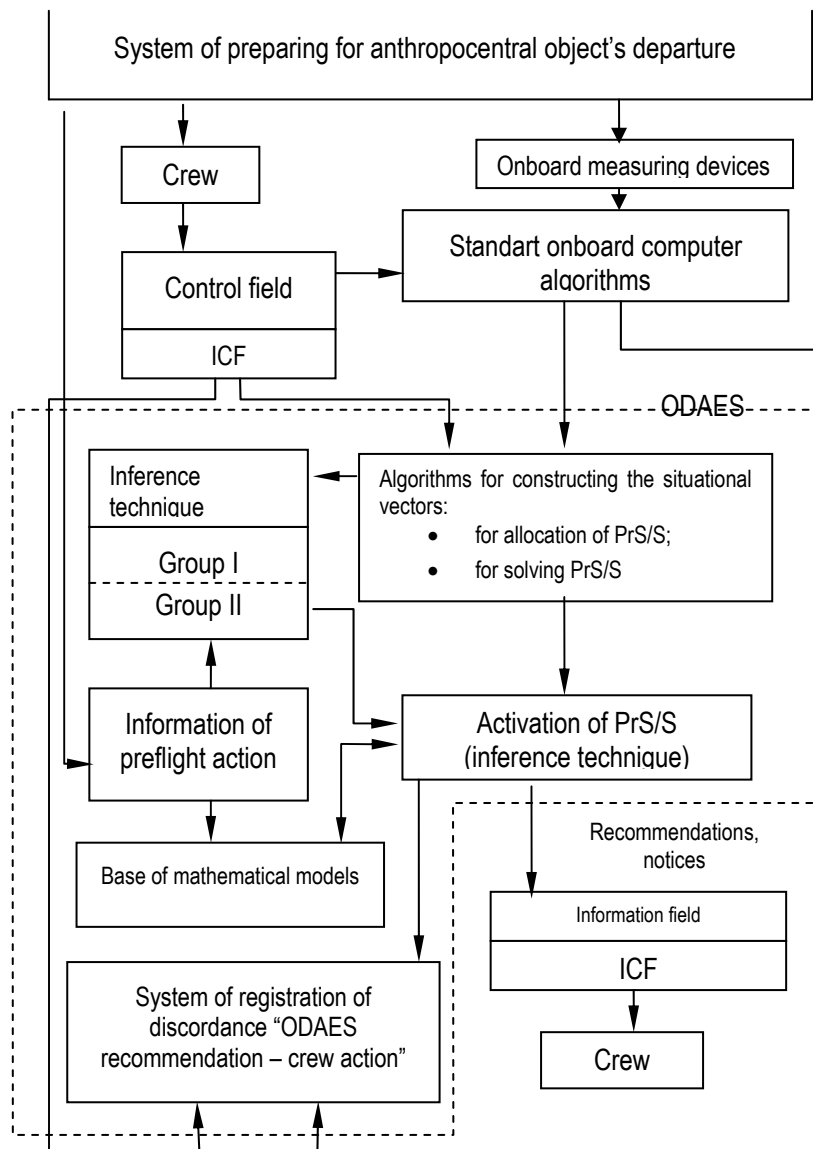


Fig. 1. The ORTAES of typical situation. Function units.

The ORTAES recommendations must be in constant agreement with the activated model of the crew behavior at the conceptual and operative levels. The crew perceives the external world where the anthropocentric object operates, and the object technical state itself (the inboard world) through the sense organs and through the crew compartment information model of the outboard and inboard situation in the information-control field (the observable external world). With professional skills and prior information (the most general information about the external world and specific information about the anthropocentric object operation session), the crew (within its activated model of behavior in this TS) each time selects the current problem sub-situation (PSS) without reporting about this to the TS ORTAES. Precisely with respect to this PSS, the TS ORTAES must immediately elaborate a reasonable and efficient recommendation for solving it.

Let us enumerate the main features of a TS ORTAES:

- it must solve all the problems of its own typical situation;
- it must have a restricted dialogue with the crew (restrictions on the time interval allowed by the external situation and on the information input through the information-control field of the anthropocentric object crew compartment performed by an operator);

- as for algorithms and rules, it must be oriented to the situational control structures;
- it must be in constant agreement with the activated conceptual model of operator behavior by generating recommendations for solving the current problem at the level of a professional operator with a significance that is sufficient for him (so as not to convey information the operator already probably knows);
- it must have a delayed self-education component.

---

## 2. Structure of The Knowledge Base of The TS ORTAES

---

The TS ORTAES always operates (in the real-time mode) in cooperation with an operator [Fedunov, 1996]. It constantly produces and presents recommendations for the operator about a rational method for solving the current problem (PrS/S). In this context, the operator within the activated TS does not report to the ORTAES about the problem that has occurred and about the necessity to present recommendations for its solution to him. Furthermore, the operator can disregard (without reporting to the ORTAES) the recommendation elaborated by the ORTAES, and the ORTAES has to operate further by taking into account the operator's decision (made contrary to its recommendation).

We briefly dwell on the description of the destination and the form of the inference technique in the knowledge base of an ORTAES (Fig. 1).

Using the current information from onboard measuring devices, standard algorithms in onboard computer, signals from the information-control field (ICF) of the crew compartment, a situation vector SV(TS-PrS/S) is formed in the knowledge base of the ORTAES. This vector describes the state of the outboard and onboard environment for assigning (or identifying) current PrS/S. We call the technique of such an assignment the inference technique on the set of PrS/S. It is constructed on the basis of results of cooperating with experts who are specialists in the object domain considered. These mechanisms are implemented in the ORTAES in the form of the rules "if ..., then ..., else...". Their completeness and consistency is achieved by finalizing the ORTAES on systems of imitational modeling (SIM) [Fedunov, 1996] together with experts.

The inference technique used for (determining) a rational solution to the current PrS/S is represented by three types of mechanisms.

### 2.1. Inference Technique Taking into Account the Current Preferences of the Crew.

Let a problem and some alternatives for its resolution be given. Suppose that, in each alternative, we are interested in some of its properties, which we will use for comparing the alternatives while choosing the most preferable one. We shall call these properties the comparison criteria. Suppose that we have several criteria which we use for comparing the alternatives. Also sup: that there is an expert (or experts) who has a sufficiently definite opinion about the problem and the alternatives for solving this problem, which allows him to pairwise compare the alternatives according to each criterion.

The method of multicriteria choice of an alternative is a systematic procedure for hierarchical ordering of the elements of the problem. This method allows one to arrange the alternative according to their preferences with respect to a totality of specified comparison criteria. In order to constructively use this method in the inference technique, we present its justification based on studies by Saaty [Saaty, 1991].

#### 2.1.1. A Method of Pairwise Comparison of Alternatives with one Preference Criterion.

Let us distinguish one of the comparison criteria and pairwise compare the alternatives with respect to this criterion.

To formalize the procedure for choosing a preferable alternative, first, we consider the simplest example, namely, the problem of choosing the heaviest object (the comparison criterion) in a given set of objects  $A_1, \dots, A_i, \dots, A_n$  (alternatives) whose absolute weights (physical) are known. The weight of the object  $A_1$  is  $w_1$ , the weight of the object  $A_2$  is  $w_2$ , and the weight of the object  $A_n$  is  $w_n$ . Since the object weights are known, to range them by weight, it is sufficient to arrange their weights in ascending order (to sort  $n$  numbers) and to choose the object with the greatest weight. Let us see in what ways one can choose the heaviest object.

We shall pairwise compare the weights of the objects  $A_i$ ;  $i = 1 - n$ , recording the comparison results in the form of a table (matrix).

The properties of an ideal matrix of pairwise comparisons can also be used for experimental matrices.

Let a researcher have a sufficient amount of qualitative information about some instances  $A_1, \dots, A_n$  compared by a certain criterion. Suppose that the matrix of pairwise comparisons of order  $n$  (experimental matrix) is composed for these instances. Naturally, it differs from the ideal matrix (calculated for this case). To estimate this difference, Saaty proposes the following procedure [Saaty, 1991].

Calculate the consistency index (CI) of the experimental matrix  $CI = \frac{\lambda_{\max} - n}{n - 1}$ , where  $\lambda_{\max}$  is the maximal eigenvalue of the experimental matrix of pairwise comparisons and  $n$  is the order of this matrix.

One can immediately see that, if the experimental matrix is ideal, then  $CI = 0$ .

In [Saaty, 1991], an estimate (the random consistency index) is introduced for an arbitrary square matrix of order  $n$ , which is positive, inversely symmetric, and has a unit principal diagonal. In the same paper, a table of the random consistency indices (RCI) of such matrices is given. A fragment of the table from [Saaty, 1991] is presented in Table 1.

Finally, Saaty proposes to calculate the consistency ratio (CR):  $CR = \frac{CI}{RCI}$ , where RCI is founded from

table 1. For  $0 \leq CR \leq 0.10-0.15$ , he proposes to consider the experimental matrix as close to the ideal one and to use all useful properties of the latter for the experimental matrix. The main point of these properties is that, by the eigenvector of the experimental matrix corresponding to its maximal eigenvalue  $\lambda_{\max}$ , one can judge the priorities of the instances compared according to the criterion considered.

Table 1

Order of the matrix, n	3	4	5	6	7	8	9	10
RCI	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

It is known that the determination of the eigenvalues and the corresponding eigenvectors for matrices of a high order  $n$  is a fairly complicated problem. We propose an approximate method for determining them. This method is based on the properties of the ideal matrix of pairwise comparisons.

Let us use these properties of the ideal matrix of pairwise comparisons for arranging the arbitrary alternatives (instances)  $A_1, \dots, A_i, \dots, A_n$  according to a certain criterion. Suppose that the matrix of pairwise comparisons of these alternatives is composed, and suppose that, checking its consistency, we find that this matrix is reasonably consistent; i.e.,  $0 \leq CR < 0.10-0.15$  for this matrix.

Then, the following Rule 1 is valid.

**Rule 1.** For a reasonably consistent experimental matrix of pairwise comparisons of the alternatives  $A_1, \dots, A_i, \dots, A_n$  with respect to the criterion  $K_j$ , the vector of priorities is determined in the an approximate method.

**Rule 2.** The vector of priorities can be used if the experimental matrix of pairwise comparisons is reasonably consistent.

### 2.1.2. Constructing the Matrices of Pairwise Comparisons (Experimental Matrices) in Practical Problems.

We have considered the case where the "exact" weight ratios are presented in the matrix of pairwise comparisons.

However, in practical problems, it is often impossible to exactly measure the results of pairwise comparisons. One of the methods for quantitatively estimating the ratio of alternatives (for the ideal matrix, this is the ratio of the object weights) is the use of a numeral scale.

The method used most often is the Saaty 9-mark scale. If one uses the Saaty scale, then the matrix of pairwise comparisons is inversely symmetric.

It is convenient to represent the results of pairwise comparisons in the form of matrix. In the upper left corner of the matrix, we write the name of the criterion, according to which the objects are pairwise compared.

In the example considered above, it is the weight. The rows and columns of the matrix correspond to the names of alternatives. The vertical ordering of the names (the first column of the matrix) and the horizontal one (the first row of the matrix) are the same. Any cell of the matrix contains the result of the pairwise comparison of the alternative in the row with the alternative in the column. This result is estimated according to the Saaty scale.

### 2.1.3. Multicriteria Choice of an Alternative.

Let there be several alternatives  $A_1, \dots, A_i, \dots, A_n$  for the solution to a problem. These alternatives should be ordered according to criteria  $K_1, \dots, K_j, \dots, K_s$ .

For any criterion  $K_j$ , we estimate the weights of the alternatives  $A_1, \dots, A_i, \dots, A_n$

$$S(K_j) = \{S_1(K_j), \dots, S_i(K_j), \dots, S_n(K_j)\}$$

Using the method of pairwise comparisons and estimating the results by the Saaty scale, we determine the weights of the criterion significances  $S = \{S_1, \dots, S_j, \dots, S_s\}$  for the researcher.

Then, for any  $A_i$ , it is natural that its weight according to a criterion  $S_i(K_j)$  is taken into account in the resulting weight for all criteria with the coefficient equal to the weight of this criterion's significance. The total weight (priority, rating) of the  $i$ -th object is determined by the formula

$$R_i = S_1(K_1) S_1 + \dots + S_i(K_j) S_j + \dots + S_s(K_s) S_s = \sum_{j=1}^s S_i(K_j) S_j$$

Finally, the alternatives  $A_1, \dots, A_i, \dots, A_n$  in the problem of multicriteria choice are arranged in accordance with the total weights. The alternative with the greatest total weight is the most preferable according to the whole set of comparison criteria.

### 2.1.4. The Structure of the Inference Technique in ORTAES Constructed on the Basis of the Algorithm of Multicriteria Choice.

The knowledge base of ORTAES contains a mathematical model for generating alternative versions for resolving problem sub-situations of admissible types which are fed into ORTAES at the stage of preparation for the operation session of the anthropocentric object (for piloted aircraft, when preparing for departure). The MM contains algorithms for determining the criterion values  $K_j \in \{K_j\}$  for any alternative generated.

Current information characterizing the problem sub-situation and admissible types of alternatives for resolving this PrS/S is supplied at the input of the MM. On the basis of admissible types of alternatives and the existing conditions for the occurrence of the PrS/S, in the MM, a complete set of alternatives  $\{A_i\}$  of admissible types is generated, and, for any alternative  $A_i \in \{A_i\}$ , the numerical value of each criterion  $K_j \in \{K_j\}$  is calculated. Moreover, an operative correction (by the crew or onboard computer) of the values of some coordinates of the vector SV(PrS/S-solution) characterizing the PrS/S is possible.

Thus, any alternative (from the set generated by the MM) is characterized by a vector whose coordinates are the numerical values of the criteria  $K_j$ .

On the basis of these vectors, the matrices of pairwise comparisons of the alternatives are constructed for each criterion.

We separately dwell on the matrix of pairwise comparisons of criteria. When constructing this matrix, one should maximally take into account the crew preferences formed by analyzing the existing current (for the operation session) situation. Taking into account that the crew's possibility of inputting this information is unlikely, one should maximally use the transitivity property of the matrix of pairwise comparisons when constructing this matrix.

After this, the vector of total weights of alternatives is calculated by the algorithm presented in Subsection 2.1.4. An example of implementing the inference technique described is given in [Musarev and others, 2001].

## 2.2. Inference Technique Based on Precedents.

Such inference methods are used in problem subsituations, whose complexity does not allow one to constructively formalize them, but for which there is some experience (precedents) of their successful resolution.

One of difficulties of this approach is the correct choice of the coordinates  $(x_1, \dots, x_i, \dots, x_n)$  of the situational vector SV(PrS/S-solution), both in their number and in the form of representation of each coordinate. The completeness

of the description of the situational vector and the connection of a particular vector with a particular precedent is established by long-term cooperation with experts, who are actual bearers of this knowledge.

As a rule, the coordinates of the situational vector are linguistic variables.

**2.2.1. Linguistic Variable as a Coordinate of the Situational Vector.**

A linguistic variable is defined by Zadeh in [Zadeh,1976] as a variable whose values belong to a specified set of terms or expressions of a natural language. The latter were also called terms.

To work with linguistic variables, one should represent each term via an appropriate fuzzy set [Zadeh,1976, Rotshtein,1999]. The latter, in turn, is represented via a universal set (universe) and the membership function of the elements of the universal set to the considered fuzzy set. The membership function takes the values in the interval [0, 1]. It quantitatively estimates the grade of membership of an element in a fuzzy set.

Note that both the universal sets and the membership functions on the set are specified on the basis of investigation results (together with experts) of the corresponding object domain.

For a large number of terms, their membership functions are usually specified in a unified form. Most often, this is a piecewise linear function.

**2.2.2. Knowledge Matrices by Precedents.**

Let a state of a problem sub-situation be described by a situational vector with coordinates  $(x_1, \dots, x_i, \dots, x_n)$  and each coordinate  $x_i$  be a linguistic variable with a set of terms  $A_i = \{a_i^1, \dots, a_i^j, \dots, a_i^{K_i}\}$ . For certain realizations of the situational vector, where each linguistic variable takes one of its possible values (a concrete term), there is a precedent of successful resolution of this PrS/S.

Suppose that a set  $d_j, j = 1, \dots, p$ , of precedents is accumulated and each precedent is associated with a set of particular situational vectors, for which this precedent has been selected.

Table 2

Nos.	Coordinates of situational vector					Precedent
	$x_1$	...	$x_i$	...	$x_n$	
1.1	$a_1^{11}$	.....	$a_i^{11}$	.....	$a_n^{11}$	$d_1$
:	:	:	:	:	:	
$1K_1$	$a_1^{1K_1}$	.....	$a_i^{1K_1}$	.....	$a_n^{1K_1}$	
:	:	:	:	:	:	:
p1	$a_1^{p1}$	.....	$a_i^{p1}$	.....	$a_n^{p1}$	$d_p$
:	:	:	:	:	:	
$pK_p$	$a_1^{pK_p}$	.....	$a_i^{pK_p}$	.....	$a_n^{pK_p}$	

Let us construct the matrix of this correspondence (Table 2). We select the rows of the matrix corresponding to a precedent (the block of the precedent). Any row of the matrix is a concrete situational vector for which the corresponding precedent has been successfully realized in the past.

We enumerate the rows of the block of precedent  $d_j$ , with two indices: the first index is the number of the precedent (here, it is the number of the block), and the second index is the serial number of the situational vector in this block.

This matrix determines a system of logical propositions of the form "if..., then..., else..." For instance, the row  $j_1$  of the matrix encodes the following proposition:

$$\text{if } x_1 = a_1^{j_1} \text{ and } x_2 = a_2^{j_1} \text{ and } \dots \text{ and } x_i = a_i^{j_1} \text{ and } \dots \text{ and } x_n = a_n^{j_1}, \text{ then } d_j, \quad (2.1)$$

else a similar proposition for the next row, etc. The obtained system of logical propositions ordered in this way is called a fuzzy knowledge matrix or, simply, a knowledge matrix.

**2.2.3. Algorithm for Calculating the Membership Function of Precedent  $d_j$ .**

First of all, we present an algorithm [Zadeh,1976] for determining the membership function  $\mu_{d_j}(x_1, .. x_i, .. x_n)$  of the precedent  $d_j$  interpreted as a fuzzy set on a universal set  $U_d = U_{x_1} \times ... \times U_{x_i} \times ... \times U_{x_n}$ , where  $U_{x_i}$  is a universal set on which the terms of the linguist variable  $x_i$  are defined, and  $U_d$  is the Cartesian product of the universal sets  $U_{x_i}$ .

Any logical proposition of the type (2.1) or, equivalently, any row of the knowledge matrix is a fuzzy relation of the corresponding fuzzy sets. For instance, for (2.1), this is  $a_1^{j1} \times a_2^{j2} \times ... \times a_n^{jn}$ .

In accordance with [6, 7], the membership function of a fuzzy set generated by this fuzzy relation is  $\mu_{a_1^{j1}}(x_1) \wedge ... \wedge \mu_{a_i^{ji}}(x_i) \wedge ... \wedge \mu_{a_n^{jn}}(x_n)$ , where " $\wedge$ " we denote the "min" operation.

Analyzing the whole block of logical propositions related with precedent  $d_j$  (the block of the corresponding rows of the knowledge matrix), note that they form the union of the corresponding fuzzy sets generated while considering the rows of the selected block. In accordance with [6, 7], the membership function of this union, which is identified with the membership function of the precedent  $d_j$ , is

$$\mu_{d_j}(x_1, \dots, x_i, \dots, x_n) = (\mu_{a_1^{j1}}(x_1) \wedge \dots \wedge \mu_{a_i^{ji}}(x_i) \wedge \dots \wedge \mu_{a_n^{jn}}(x_n)) \vee \dots \vee (\mu_{a_1^{jk_j}}(x_1) \wedge \dots \wedge \mu_{a_i^{jk_j}}(x_i) \wedge \dots \wedge \mu_{a_n^{jk_j}}(x_n))$$

where by " $\vee$ " we denote the "max" operation.

Table 3.

Nos	Coordinates of the situational vector					min	max	d
	$x_1$		$x_i$		$x_n$			
:	:	:	:	:	:	:	:	:
$j_1$	$(a_1^{j1})^*$	.....	$(a_i^{j1})^*$	.....	$(a_n^{j1})^*$	$\min_i (a_i^{j1})^*$	$\max_{j_s} \min_i (a_i^{j_s})^*$	$\mu_{d_j}$
:	:	:	:	:	:	...		
$j_s$	$(a_1^{j_s})^*$	.....	$(a_i^{j_s})^*$	.....	$(a_n^{j_s})^*$	$\min_i (a_i^{j_s})^*$		
:	:	:	:	:	:	...		
$j_{K_j}$	$(a_1^{j_{K_j}})^*$	.....	$(a_i^{j_{K_j}})^*$	.....	$(a_n^{j_{K_j}})^*$	$\min_i (a_i^{j_{K_j}})^*$		

Formally, this algorithm for determining the membership function of the precedent  $d_j$  can be written in the following form:

- (a). fix an arbitrary point  $(x_1^*, \dots, x_i^*, \dots, x_n^*) \in U_{x_1} \times \dots \times U_{x_i} \times \dots \times U_{x_n}$  ;
- (b). for any block of the knowledge matrix corresponding to  $d_j$  determine  $\mu_{d_j}(x_1, .. x_i, .. x_n)$  at this point according to the scheme of Table 3.

Note that, for any fixed point  $(x_1^*, x_i^*, \dots, x_n^*)$ , the block of the matrix presented in Table 3 is numerical, because each term  $a_i^{j_s}$  from this block is replaced with the value of its membership function  $(a_i^{j_s})^*$  calculated at the corresponding  $x_i^*$ . The operation  $\min_i a_i^{j_s}$  is performed with the numbers located in rows "i"  $1 \leq i \leq n$ ,

and the minimal number in the corresponding row is placed in the column "min." The operation  $\max_{j \in J} \min_i a_i^{jS}$  selects the greatest of the row minima obtained for  $1 \leq j_s \leq K_j$ . This number is the value of the membership function  $\mu_{dj}(x_1, \dots, x_n)$  at the fixed point  $(x_1^*, \dots, x_i^*, \dots, x_n^*)$ . Performing this calculation for every point of the universal set, we obtain the membership functions that interest us.

#### 2.2.4. Algorithm for Choosing a Precedent when Observing a Situational Vector with Quantitative Coordinates.

When observing a situational vector [Rotshtein, 1999] with quantitative coordinates (all coordinates of the vector are measured by numerical scales), in order to select the most preferable precedent, it is not necessary to completely determine the membership functions  $\mu_{dj}(x_1, \dots, x_i, \dots, x_n)$  on the whole set of points of the universal set. It is sufficient to calculate their values only for fixed numerical values of the coordinates of the vector, which is obtained by us as a result of the observation. For this purpose, we should use the algorithm from Subsection 3.3 once, taking the coordinates of the observed situational vector as  $(x_1^*, \dots, x_i^*, \dots, x_n^*)$ .

As a result, for any precedent  $d_j$ , we obtain a number  $d_j(x_1^*, \dots, x_i^*, \dots, x_n^*)$ , which is the grade of membership of  $d_j$  to the point  $(x_1^*, \dots, x_i^*, \dots, x_n^*)$ .

Starting from this interpretation, the most preferable precedent for resolving the observed PrS/S is the precedent  $d_j^*$  such that

$$d_j^*(x_1^*, \dots, x_i^*, \dots, x_n^*) = \max_{1 \leq j \leq p} d_j(x_1^*, \dots, x_i^*, \dots, x_n^*).$$

#### 2.3. Three types of inference techniques are presented.

They are constructive for resolving problem substitutions of typical situations of anthropocentric object functioning (for instance, flights of piloted aircraft).

The first type of inference technique (on product rules) is based on results of the mathematical investigation of the PrS/S and, to a certain degree, on the knowledge of experts in the object domain under consideration. This technique is widely used by designers of the knowledge bases of the first Russian and foreign ORTAES.

The second type of inference technique is based on the use of the algorithm for multicriteria choice of an alternative. The technique is directed to the substantial use of the results of the preparation for the operation session of the anthropocentric object (prior information) and crew preferences formed while operatively analyzing the situation existing in the current operation session. The considered example of the use of this technique allows one to hope that it can be successfully employed in practice when designing the real knowledge bases of ORTAES.

The third type of inference technique has been studied only theoretically and has not been tested on practical examples. It is directed to be used in the inference technique of precedents and is based on the algorithms for choosing a solution on the basis of the knowledge matrix. These algorithms are successfully applied in diagnostic problems.

The choice of the type of inference technique for constructing in ORTAES a recommendation for resolving a problem sub-situation of a concrete type depends on its complexity and on the possibility of adequately formalizing it mathematically.

---

## Conclusion

(1) The TS ORTAES of anthropocentric objects is a class of dynamic intelligent systems presenting, in the real-time mode, to the crew of an anthropocentric object the complete recommendations on its rational behavior under current operation conditions.

The TS ORTAES has the following specific features:

- 
- working in the real-time mode of operation of the anthropocentric object that carries the TS ORTAES with the restricted dialogue "TS ORTAES-crew" under continuous coordination of ORTAES recommendations and activated (at the current instant) conceptual model of the crew behavior;
  - solving all problems of the TS for which the TS ORTAES is developed;
  - the procedure of the recommendation inference in the TS ORTAES is based on the principles of situational control;
  - the delayed component of the TS ORTAES self-education.
- (2) The semantic structure of the TS ORTAES knowledge base consists of the following:
- a two-level (with respect to semantics) rule base identifying, at the upper hierarchy level, the problem sub-situation (PSS) that occurred and determining methods for its solution at the lower hierarchy level;
  - mathematical model (MM) bases consisting of MMs of tree types (the MMs predicting space-time occurrence of significant events, the MMs generating admissible solutions and their ranking, and the MMs for estimating unobservable phase coordinates);
  - three types of inference techniques are presented. They are constructive for resolving problem sub situations of typical situations of anthropocentric object functioning.
- 

### Bibliography

---

- [Fedunov,1998] Fedunov B.E. Constructive Semantics of Anthropocentric Systems for Development and Analysis of Specifications for Onboard Intelligent Systems, *Izv. Ross. Akad. Nauk, Tear. Sist. Upr* (Simultaneous English language translation of the journal is available from MAIK "Nauka/Interperiodica" (Russia). *Journal of Computer and Systems Sciences International* ISSN 1064-2307), 1998, № 5.
- [Fedunov,1996] Fedunov B.E., Problems of the Development of Onboard Operative and Advising Expert Systems for Anthropocentric Objects, *Izv. Ross. Akad. Nauk, Tear. Sist. Upr.*, (Simultaneous English language translation of the journal is available from MAIK "Nauka/Interperiodica" (Russia). *Journal of Computer and Systems Sciences International* ISSN 1064-2307), 1996, № 5.
- [Vasil'ev and others,2000] Vasil'ev S.N., Gherlov A.K., Fedosov E.A., Fedunov B.E. Intelligent control of dynamic systems. Moscow, Fizmatlit, 2000, (by Russian).
- [Musarev and others,2001] Musarev L.M. and Fedunov B.E., The Structure of Target Allocation Algorithms on Board the Commander of an Aircraft Group, *Izv. Ross.Akad. Nauk, Teor. Sist. Upr.* (Simultaneous English language translation of the journal is available from MAIK "Nauka/Interperiodica" (Russia). *Journal of Computer and Systems Sciences International* ISSN 1064-2307),,2001, № 6.
- [Saaty,1991] Saaty T. The Analytic Hierarchy Process. Planning ,Priority Setting, Resource Allocation, New York: McGraw-Hill, 1980. Translated under the title *Analiticheskoe planirovanie. Organizatsiya sistem*, Mocosw:Radio i Svyaz', 1991.
- [Zadeh,1976] Zadeh L. The Concept of a Linguistic Variable and Its Application to Approximate Reasonong, New York:Elsevier, 1973. Translated under the title *Ponyatie lingvisticheskoi peremennoi i ee primenenie k prinyatiyupriblizhennykh reshenii*, Moscow: Mir, 1976.
- Rotshtein,1999] Rotshtein A.P. *Intellectual'nye tekhnologii identifikatsii* (Intelligent Technologies of Verification), Vinnitsa: Universum, 1999.
- 

### Author's Information

---

Boris Evgenjevich Fedunov – State Scientific Research Institute for Aviation System (GosNIIAS), ul. Viktoenko 7, Moscow, 125319 Russia; e-mail [boris\\_fed@gosniias.ru](mailto:boris_fed@gosniias.ru)



---

## DEFINING NETWORK ACTIVITY PATTERNS USING FIRST ORDER TEMPORAL LOGICS

Lubomir Stanchev

*Abstract:* Part of network management is collecting information about the activities that go on around a distributed system and analyzing it in real time, at a deferred moment, or both. The reason such information may be stored in log files and analyzed later is to data-mine it so that interesting, unusual, or abnormal patterns can be discovered. In this paper we propose defining patterns in network activity logs using a dialect of First Order Temporal Logics (FOTL), called First Order Temporal Logic with Duration Constrains (FOTLDC). This logic is powerful enough to describe most network activity patterns because it can handle both causal and temporal correlations. Existing results for data-mining patterns with similar structure give us the confidence that discovering DFOTL patterns in network activity logs can be done efficiently.

*Keywords:* network management, temporal logics

*ACM Classification Keywords:* C.2.3 Network Operations – network management; F.4.1 Mathematical Logic – temporal logic

---

### 1. Introduction

The rapid growth of the Internet has increased the need for recording and analyzing network activity patterns. The vast amount of information that is collected makes its manual processing unfeasible. Fortunately, data-mining algorithms have been developed for the automatic and semi-automatic processing of big chunks of data. In this paper we explore how useful information can be extracted from network activity logs in an efficient way through the use of semi-automatic data exploration procedures.

Network activity logs are important for network management. Related research (e.g., [LeFa93]) shows that log analyzing activities can be helpful for all five categories of network management. For example, fault management network logs can be analyzed to derive the patterns that correlate alarms with actual faults. Such patterns can then be used by the network management station to feed to the network administrator the faults that actually occurred, rather than their symptoms. In configuration management, the network log can be used to deduce what system components are active, how they are configured, and how they are connected. In security management, historical logs can be analyzed in order to discover patterns of irregularities that can point to a hacker attempt to break in the system or to an actual breach of security. In performance management, patterns that are data-mined from log files can show the network administrators which system components are under-utilized and which are over-utilized so that appropriate adjustments can be made. Finally, in accounting management, patterns extracted from network logs can help the system administrators get a better understanding of which groups of users rely on which network resources. Serious abuse of the system can also be detected from analyzing network logs.

There are two known types of procedures for acquiring network activity information. The first method presumes that the network management station has traps set at different locations throughout the system. An example of a trap for a network link is: "If the traffic over the link reaches 90% of the link capacity, then send an alarm". An example of a practical implementation of this approach is Remote Monitoring (RMON) with Simpler Network Management Protocol (SNMP) (see [GaTa99]). The second approach assumes that the network management station will periodically poll managed objects or network monitoring components. An example of a system which uses both techniques is Network Flight Recorders (NFR) - see [WeChSt99].

The first contribution of this paper is the introduction of a dialect of First Order Temporal Logics (FOTL) that we call *First Order Temporal Logics with Duration Constraints (FOTLDC)*. This dialect can be used to express patterns in network activity logs. FOTLDC is more expressive than many existing languages for network activity pattern definition (e.g., [HaSuVi99], [MaToVe95],[LePaSt99], and [LePaSt99]). Unlike the enumerated languages,

FOTLDC can be used to express not only *causal*, but also *temporal* patterns. It also allows for the expression of patterns that reference attribute values of managed objects over time periods, which is not supported even in rich temporal-based models, such as the one described in [LiMoYa99]. To summarize, FOTLDC can capture useful patterns in network activity logs that cannot be captured by existing proposals.

The second contribution of this paper is the architectural design of a system that analyzes network activity logs based on FOTLDC patterns. The most challenging part of implementing such a system is detecting which patterns are normal and which are abnormal. Part of this information can be provided by experts and inferences from this expert knowledge can also be used. However, a good log-analyzer system should be able to classify on its own the patterns as normal and abnormal. This can be done by observing the frequency of the different events in a network activity log and classifying the more frequently occurring patterns as normal and the less frequently occurring patterns as abnormal. The actual thresholds can be set by a domain expert. Note that, in order for this approach to work, the data in the network activity log must be representative. For example, if the data that is analyzed is taken during a hacker's attack, then the data-mining engine may deduce that patterns of unauthorized network intrusion are normal.

To summarize, we propose the creation of a process for analyzing network activity logs based on discovering FOTLDC patterns. The novelties in our paper are:

- extending first order temporal logics with duration constraints,
- representing network activity patterns as temporal logic formulas, and
- presenting an architectural design of a system that analyzes network activity logs using the proposed temporal logic.

In what follows, in Section 2 we define the syntax and semantics of the FOTLDC. We also explore how this logic is different from existing temporal logics and why it is suitable for describing network log patterns. Next, in Section 3 we present existing data-mining approaches for discovering causal and temporal patterns and discuss how they can be applied to FOTLDC patterns. In Section 4, we describe the network activity log analyzing process. Section 5 contains a brief summary of the paper and direction for future research.

---

## 2. First Order Temporal Logic with Duration Constraints

---

We next present the syntax of FOTLDC. This logic is an extension of the temporal logic proposed in [Chom95] with duration constraints. The elements of a formula in FOTLDC are:

- a finite set of predicate symbols  $p_i$  over  $\Delta$ ,
- the logical connectives  $\wedge, \vee, \neg$ , and  $\Rightarrow$ ,
- a countable infinite set of variables  $V$  over  $\Delta$ ,
- a finite set of constant symbols over  $\Delta$ ,
- the operations  $+, -, *, /$ ,
- the equality and inequality symbols:  $=, \neq, >, <, \geq$ , and  $\leq$ ,
- the quantifiers  $\forall$  and  $\exists$ , and
- the past and future temporal connectives  $\blacklozenge_k, \blacktriangleright_k, \blacksquare_k$ , and  $\square_k$ , where  $k \in T \cup \{\infty\}$ .

In above definition,  $T$  is the time domain, which we assume to be isomorphic to the set of natural numbers, and  $\Delta$  is the domain of discourse. The semantic of a FOTLDC formula is defined relative to a time instance  $t$  and a finite-time temporal database  $D = (D_0, D_1, \dots, D_k)$ . Such a database represents the different instances of a database through time, where  $D_t$  denotes the database state at time instance  $t$  (for a formal definition of a temporal database see [ChDa98]). A valuation  $v$  that interprets a FOTLDC formula is defined as a mapping from  $V$  to  $\Delta$ , where  $V$  is the set of variables. Note that we assume that value of a variable does not change with time. Informally, the meaning of the newly introduced connective  $\blacklozenge_k (\blacktriangleright_k)$  is: in at least one point of time in the past (future)  $k$  time units the formula that follows was (will be) true. The definition is relative to the current time moment  $t$ . Similarly, the connective  $\blacksquare_k (\square_k)$  means that always in the past  $k$  time units (always in the next  $k$  time units), relative to the current time moment  $t$ , the formula that follows was (will be) true. A formal definition follows.

$$D, v, t \models \blacklozenge_k \phi \text{ iff } \exists t' \in T \text{ such that } 0 < t - t' \leq k \text{ and } D, v, t' \models \phi$$

$$D, v, t \models \blacklozenge_k \phi \text{ iff } \exists t' \in T \text{ such that } 0 < t' - t \leq k \text{ and } D, v, t' \models \phi$$

$$D, v, t \models \blacksquare_k \phi \text{ iff } \forall t' \in T \text{ such that } 0 < t - t' \leq k, \text{ it is the case that } D, v, t' \models \phi$$

$$D, v, t \models \square_k \phi \text{ iff } \forall t' \in T \text{ such that } 0 < t' - t \leq k, \text{ it is the case that } D, v, t' \models \phi$$

In order to define patterns with FOTLDC, we need to represent the network activity logs as finite-time temporal databases. Fortunately, temporal databases are a natural way of representing data that is collected as a result of polling managed objects periodically. Data collected in the management station as a result of alarms, for example, SNMP traps, can also be represented in a temporal fashion.

We next present an example of a FOTLDC pattern for network activity. Suppose we want to define the pattern that if  $host_1$  goes down and stays down for more than two minutes, then the utilization of  $link_1$  will be reduced to below fifty percent in the next five minutes. This can be expressed using the following FOTLDC expression.

$$\square_{120} [\text{host\_status}(host_1, \text{off})] \Rightarrow \blacklozenge_{300} [(\exists Z, \exists W) (\text{link\_throughput}(link_1, Z, W) \wedge (z/w < 0.5))]$$

Note that time is represented with granularity of one second. The predicates `host_status` corresponds to the temporal relations `host_status`, which holds exactly when the host specified by the first argument has the state specified by the second argument. The predicate `link_throughput` correspond to the temporal relations `link_throughput`, which holds exactly when the link specified by the first attribute has the link usage specified by the second attribute and the capacity specified by the third attribute. In order to determine if the example FOTLDC expression is true, one has to check that the FOTLDC formula holds for any time instance  $t$  for which there exists a log entry.

---

### 3. Data Mining Temporal Data

---

The activity logs for a distributed system contain huge amount of data about the different activities over time. Useful information from such logs can be extracted, for example, using data mining techniques such as the ones described in [AgImSw93]. Note however that the goal of rule extraction is to extract rules of the form:  $X \rightarrow Y$ , where  $X$  and  $Y$  are first order formulas over a database. For example, when applied to a supermarket database, a data-mining algorithm may discover that people who purchase dippers also purchase beer in eighty percent of the cases. In data-mining literature, this kind of dependency is usually referred to as a *casual* dependency. Although causal dependencies are important in analyzing network activity logs, they are not powerful enough to describe all interesting correlations between events. For example, an interesting pattern that one can hope to discover in a network log for the purpose of fault management is the occurrence of several events in a particular order in a short span of time. In order to describe such a pattern, one needs to use *temporal* dependencies.

Little research has been done in the area of data-mining temporal dependencies - [MaToVe95] and [Chom95] are two exceptions. The paper [MaToVe95] proposes an algorithm for data-mining *sequential* and *parallel* episodes from a sequence of events. A sequential episode specifies that a set of events occur in particular order in the same time window. A parallel episode specifies that a set of events occur within a time window in any order. The paper shows that any pattern of event occurrences can be broken down into a composition of sequential and serial episodes. Moreover, it shows an efficient algorithm that discovers frequent patterns that occur in windows of specified size that belong to a specified, closed with respect to the operation sub-episode, set of patterns.

The [Chom95] paper presents an efficient way for storing a temporal database. As well, it describes efficient algorithms for discovering and verifying FOTL patterns. It is our hope that the algorithms from the two papers can be combined to produce an efficient algorithm for data-mining FOTLDC formulas. Designing such an algorithm remains an area for future research.

---

### 4. The Network Activity Log Analyzing Process

---

Figure 1 shows the overall process of analyzing a network activities log file. First, the network management station collects information about the state of the network over time through SNMP traps or by directly polling managed objects. This information is then passed to a temporal database and stored there. Parallel to this

process, a domain expert enters data patterns in the pattern database that are known to be normal or that are known to be abnormal.

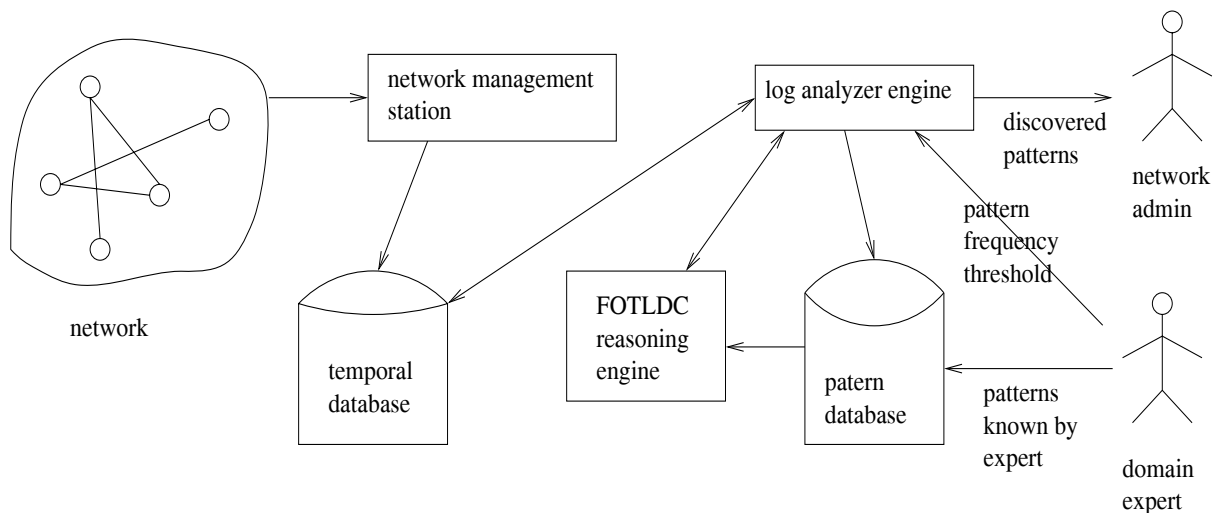


Figure 1. Network activities log analyzing process

The key part of the process is the log analyzer engine. It has two functions. The first function is to find if patterns that are known to be normal or abnormal (the FOTLDC reasoning engine can provide this information) can be discovered in the temporal database. Once a pattern is discovered, the corresponding data in the temporal database is marked as used so it is not analyzed further. As well, if the pattern that is found to be true for the temporal database is an abnormal one, then it is reported to the network administrator. The second function of the log analyzer is to discover new patterns. Once a pattern is found, the log analyzer engine checks with the FOTLDC reasoning engine to see if the pattern is indeed new, that is, the FOTLDC reasoning engine cannot classify it as normal or abnormal based on the information in the pattern database. If the pattern is indeed new, then it is classified as normal or abnormal based on its frequency. The pattern frequency threshold is set by an expert and it specifies how often a pattern has to occur in order to be classified as frequent (i.e., normal) or infrequent (i.e., abnormal). New patterns are found by data-mining the temporal database. Whenever new patterns are discovered, they are stored in the pattern database.

The FOTLDC reasoning engine is used by the log analyzer engine to determine which patterns are known to be normal and which to be abnormal. Since patterns are expressed as FOTLDC, it is also possible to reason with them. In other words, the pattern database may be thought as a knowledge database that knows which patterns are normal and which are abnormal. The FOTLDC reasoning engine can be thought as a theorem prover which can decide if there is enough information to classify a given pattern as normal or abnormal.

## 5. Summary and Future Research

In this paper we propose architecture for analyzing network activity logs using activity patterns expressed via FOTLDC. What remains to be done is the low-level design of the system. This includes:

- the design of an efficient data-mining algorithm that discovers FOTLDC patterns,
- the design of an algorithm that implements an efficient FOTLDC reasoning engine, and
- the design for the interface between the different components in Figure 1.

After these low-level designs have been performed, an actual implementation can be coded. Then, the components from Figure 1 can be connected to an existing network. The network activity log analyzing process can then be used in a test situation to collect experimental data. The data can then be analyzed in order to determine how successful the proposed system will be in a real-world situation.

Note that the proposed methodology for examining activity logs is more or less automatic - that is, it allows for little human interaction. However, experimental results have shown that such algorithms are not very successful. Therefore, it is worth exploring how human experts can participate in the different stages of pattern discovery. For example, when a new pattern is discovered, it can be examined by a domain expert before it is recorded in the pattern database.

To summarize, we propose a methodology for network activity log processing. We believe that the theoretical background behind the proposed methodology is sound and that when applied in practice the proposed network activity analyzer will produce results better than most existing network monitoring systems. However, the only way to find this for certain is to implement the proposed network activity analyzer and compare its performance and effectiveness to existing network monitoring systems.

---

## Bibliography

---

- [AgImSw93] Agrawal R., Imielinski T., Swami A., Mining association rules between sets of items in large databases. In *Proceeding of the ACM SIGMOD Conference on Management of Data*, pp. 207-216, 1993
- [ChDa98] Chomicki, J., Toman D. Temporal Logic in Information Systems. In *Logics for Databases and Information Systems*, Kluwer Academic publishers, pp. 31-70, 1998
- [Chom95] Chomicki, Jan. Efficient Checking of Temporal Integrity Constraints Using Bounded History Encoding. In *ACM Transactions on Database Systems*, Vol 20, No. 2, pp. 149-186, June 1995
- [HaSuVi99] Hasan M., Sugla B., Viswanathan R. A Conceptual Framework for Network Management Event Correlation and Filtering Systems. In *Sixth IEEE/IFIP International Conference on Integrated Network Management*, May 1999
- [LePaSt99] Lee W, Park C., Salvatore J., Automated intrusion detection using NFR: Methods and experiences. *Workshop on Intrusion Detection and Monitoring*, 1999
- [LeFa93] Leinwand A, Fang K., Network Management - A Practical Perspective. Addison Weley, 1993
- [LiMoYa99] Liu G., Mok A.K., Yang E.J., Composite Events for Network Event Correlation, In *Sixth IFIP/IEEE International Symposium on Integrated Network Management*, May 1999
- [MaToVe95] Mannila H., Toivonen H., Verkamo A., In *Proceedings of The First International Conference on Knowledge Discovery and Data Mining*, Montreal, Canada, pp. 20-21, 1995

---

## Authors' Information

---

Lubomir Stanchev - Indiana University - Purdue University Fort Wayne, 2101 E. Coliseum Blvd., Fort Wayne, IN 46805 USA e-mail: [stanchel@ipfw.edu](mailto:stanchel@ipfw.edu)

## ON THE ERROR-FREE COMPUTATION OF FAST COSINE TRANSFORM

Vassil Dimitrov, Khan Wahid

*Abstract: We extend our previous work into error-free representations of transform basis functions by presenting a novel error-free encoding scheme for the fast implementation of a Linzer-Feig Fast Cosine Transform (FCT) and its inverse. We discuss an 8x8 L-F scaled Discrete Cosine Transform where the architecture uses a new algebraic integer quantization of the 1-D radix-8 DCT that allows the separable computation of a 2-D DCT without any intermediate number representation conversions. The resulting architecture is very regular and reduces latency by 50% compared to a previous error-free design, with virtually the same hardware cost.*

*Keywords: DCT, Image Compression, Algebraic Integers, Error-Free Computation.*

*ACM Classification Keywords: I.4.2 Compression (Coding), I.1.2 Algorithms, F.2.1 Numerical Algorithms.*

## Introduction

The Discrete Cosine Transform (DCT) is the core transform of many image processing applications for reduced bandwidth image and video transmission, including the JPEG image processing standard and high performance video coding standards such as MPEG and H.263.

Because of the enormous popularity of the DCT, much research has been published on fast DCT algorithms [Arai, 1988][Duhamel, 1990][Feig, 1992][Linzer, 1991], where the effort is devoted to reducing the number of arithmetic operations used. Scaled DCT algorithms rely on a post-pointwise scaling operation which removes some of the arithmetic operations from the main transform computation. The post-scaling operations for a 2-D DCT can be delayed until the end of the transformation process. In this paper, we discuss a new algebraic-integer-mapping for the Linzer-Feig scaled-DCT [Linzer, 1991].

Several algorithms and architectures have previously been proposed to optimize both pure and scaled DCT implementations using 1-D and 2-D algebraic integer (AI) encoding of the DCT basis functions. Both single and multidimensional AI schemes have been used for low-complexity and parallel architectures [Dimitrov, 1998][Dimitrov, 2003][Wahid, 2004]. In most of these previous published encoding techniques, conversion from the AI output of each 1-D DCT computation has been required, even if the DCT is being used in a separable 2-D DCT computation. Here we introduce a new algebraic integer encoding technique which removes the need for conversion to binary at the end of the first 1-D DCT. We also extend this concept for 2-D error-free algebraic integer encoding and supply details on the computational complexity and mathematical precision required to implement the algorithm.

## Background

For a real data sequence  $x(n)$  of length  $N$ , the DCT is defined as follows:

$$F(k) = 2 \sum_{n=0}^{N-1} x(n) \cos \left[ \frac{(2n+1)k}{2N} \pi \right]; \quad 0 \leq k \leq N-1 \quad (1)$$

The Inverse DCT (IDCT) is also defined as:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \bar{F}(k) \cos \left[ \frac{(2n+1)k}{2N} \pi \right]; \quad 0 \leq n \leq N-1 \quad (2)$$

$$\text{Where, } \bar{F}(k) = \begin{cases} \frac{F(0)}{2} & k = 0 \\ F(k) & \text{otherwise} \end{cases}$$

Both DCT and IDCT are separable transformations. So the 2-D transform can be computed by first performing 1-D transforms on each row and then performing 1-D transforms on each column.

## Algebraic Integer Encoding

Algebraic integers are defined by real numbers that are roots of monic polynomials with integer coefficients

[Dedekind, 1996]. As an example, let  $\omega = e^{\frac{2\pi j}{16}}$  denote a primitive 16th root of unity over the ring of complex numbers. Then  $\omega$  is a root of the equation  $x^8 + 1 = 0$ . If  $\omega$  is adjoined to the rational numbers, then the associated ring of algebraic integers is denoted by  $\mathbf{Z}[\omega]$ . The ring  $\mathbf{Z}[\omega]$  can be regarded as consisting of polynomials in  $\omega$  of degree 7 with integer coefficients. The elements of  $\mathbf{Z}[\omega]$  are added and multiplied as polynomials, except that the rule  $\omega^8 = -1$  is used in the product to reduce the degree of powers of  $\omega$  to below 8.

For an integer,  $M$ ,  $\mathbf{Z}[\omega]_M$  is used to denote the elements of with coefficients between  $-\frac{M}{2}$  and  $\frac{M}{2}$ .

Algebraic integer quantization has been used in DSP applications for about two decades and it has been demonstrated that it can be used for extremely efficient implementation of real-valued transforms such as the Discrete Hartley Transform [Baghaie, 2001], and the Discrete Wavelet Transform [Wahid, 2003].

**Fast Cosine Transform Algorithm**

The algorithm proposed by Linzer and Feig [Linzer, 1991] is presented as a signal flow graph in Figure 1, where  $\{a_i\}$  are input elements,  $\{S_i\}$  are scaled DCT coefficients, and fixed multipliers are given by (3):

$$c_4 = \cos \frac{4\pi}{16}; t_1 = \tan \frac{\pi}{16}; t_2 = \tan \frac{2\pi}{16}; t_5 = \tan \frac{5\pi}{16} \tag{3}$$

The outlined area in Figure 1 (with a hardware cost of 10 multiplications and 10 additions) is where our new algebraic integer mapping will be used, and we will show that our mapping technique reduces the hardware complexity and also produces error-free transform outputs.

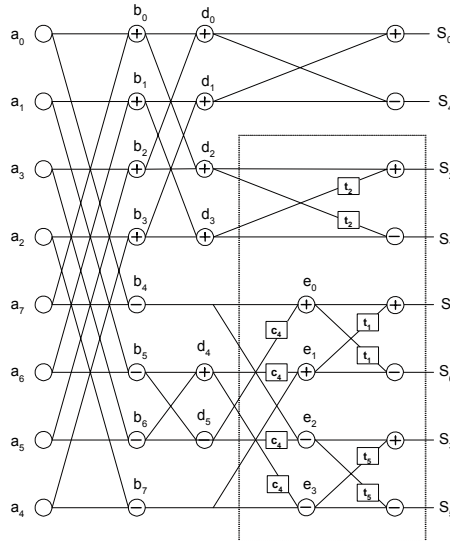


Figure 1: 1-D DCT (finite-precision binary)

1-D Algebraic Integer Encoding: Let  $z = 2 + \sqrt{2 + \sqrt{2}}$  and consider the polynomial expansion:

$$f(z) = a_0 + a_1z + a_2z^2 + a_3z^3 \tag{4}$$

Now considering the expressions:  $\cos \frac{4\pi}{16} = \frac{\sqrt{2}}{2}$ ,  $\tan \frac{\pi}{16} = \frac{2}{\sqrt{2 - \sqrt{2}}} - \sqrt{2} - 1$ ,  $\tan \frac{2\pi}{16} = \sqrt{2} - 1$ , and

$\tan \frac{5\pi}{16} = \frac{2}{\sqrt{2 + \sqrt{2}}} + \sqrt{2} - 1$ , we can represent all the multipliers from eqn. (3) exactly with integer

coefficients as shown in Table 1 (here,  $\tilde{c}_4 = 2c_4$ ).

Table 1: 1-D error-free multiplier encoding

	$a_0$	$a_1$	$a_2$	$a_3$
$\tilde{c}_4$	2	-4	1	0
$t_1$	-7	14	-7	1
$t_2$	1	-4	1	0
$t_5$	1	-12	7	-1

Note that the multiplication between any real number and the coefficients in Table 1 can now be implemented with at most 4 shifts and 1 addition (for 14). This reduces the 10 multiplications to only 5 AI additions (actually, 23 shifts and 5 additions). So, the total number of additions required to perform the first 1-D DCT is 21. In the case of an 8x8 2-D DCT, we will need a total of 736 additions including the final substitution. We note that there is no longer any precision problem since the AI encoding provides an exact representation. The flow graph of Figure 1 can now be implemented as shown simply in Figure 2.

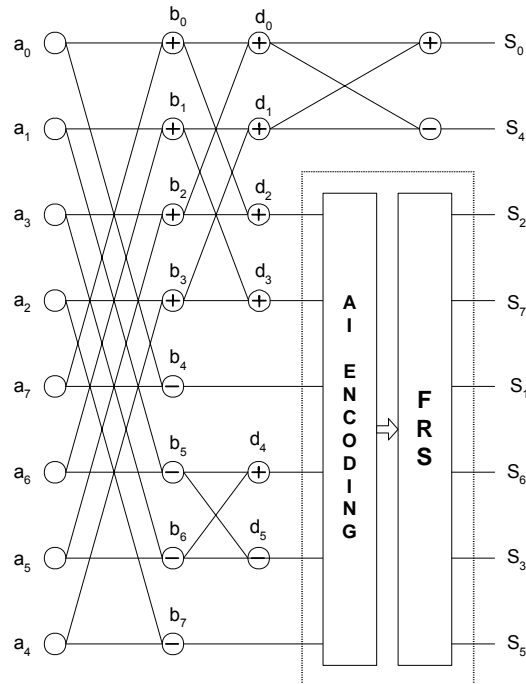


Figure 2: 1-D DCT (error-free AI encoding)

The real numbers of  $f(z)$  form a ring which may be denoted as  $Z[2 + \sqrt{2 + \sqrt{2}}]$ . Addition in this ring is component-wise and multiplication is equivalent to a polynomial multiplication modulo  $z^4 - 8z^3 + 20z^2 - 16z + 2 = 0$ , which is demonstrated in Table 2.

Table 2: Multiplication in ring  $Z[2 + \sqrt{2 + \sqrt{2}}]$

$\tilde{c}_4$	$t_1$	$\tilde{c}_4.t_1$
(2,-4,1,0)	(-7,14,-7,1)	(-8,6,-1,0)

2-D Algebraic Integer Encoding: Applying a 2-D algebraic integer scheme to this algorithm results in a more sparse representation and more flexible encoding compared to previous techniques [Dimitrov, 2003]. For this encoding, the polynomial is expanded into 2 variables:

$$f(z_1, z_2) = \sum_{i=0}^K \sum_{j=0}^L a_{ij} z_1^i z_2^j \tag{5}$$

Here, we choose K=2 and L=2 to guarantee error-free encoding. For the most efficient encoding (i.e., to obtain the sparsest matrix), we have found the following:  $z_1 = \sqrt{2 + \sqrt{2}}$  and  $z_2 = \sqrt{2 - \sqrt{2}}$ . The corresponding

multiplier coefficients (including the cross-multipliers) are encoded in the form of  $\begin{bmatrix} a_{00} & a_{10} & a_{20} \\ a_{01} & a_{11} & a_{21} \\ a_{02} & a_{12} & a_{22} \end{bmatrix}$  as shown in



Table 3. We have therefore, mapped the multiplier transcendental functions (cosine and tangent) without any error and with very low complexity. The implementation is same as shown in Figure 2.

Table 3: 2-D error-free multiplier encoding

$\tilde{c}_4$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$t_2$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$
$t_1$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$t_5$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
$\tilde{c}_4 t_1$	$\begin{bmatrix} -2 & 0 & 0 \\ 2 & -1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$	$\tilde{c}_4 t_5$	$\begin{bmatrix} 2 & -2 & 0 \\ 0 & -1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$

From Table 3, we see that the coefficients are either 0's, 1's or 2's, so no additions and at best 1 shift is required to encode these numbers which reduces the total number of AI additions to perform the first 1-D transform to only 16 (496 in total for an 8x8 2-D DCT).

**Final Reconstruction Step:** In the final reconstruction stage (FRS), we map the AI numbers to a binary output. FRS is performed based on the precision in Table 4 and the flow graphs in Figure 3 and Figure 4. For the final reconstruction, we can use Horner's rule [Knuth, 1981]. In that case, eqn. (4) and eqn. (5) can be re-written as:

$$f(z) = ((a_3z + a_2)z + a_1)z + a_0 \tag{6}$$

$$f(z_1, z_2) = a_{00} + z_1z_2(a_{11} + a_{21}z_1 + a_{12}z_2) \tag{7}$$

This final substitution stage generates some rounding errors but these errors are only introduced at the very end of the transformation process, not distributed throughout the calculation as is the case for a finite-precision binary implementation.

Table 4: FRS for different encoding scheme

Scheme	Parameter		FRS
1-D	$z = 2 + \sqrt{2 + \sqrt{2}}$	8 bits	$4 - 2^{-3} - 2^{-5}$
		12 bits	$4 - 2^{-3} - 2^{-5} + 2^{-7}$
2-D	$z_1 = \sqrt{2 + \sqrt{2}}$	8 bits	$2 - 2^{-3} - 2^{-5}$
		12 bits	$2 - 2^{-3} - 2^{-5} + 2^{-8}$
	$z_2 = \sqrt{2 - \sqrt{2}}$	8 bits	$1 - 2^{-2} + 2^{-6}$
		12 bits	$1 - 2^{-2} + 2^{-6} - 2^{-11}$

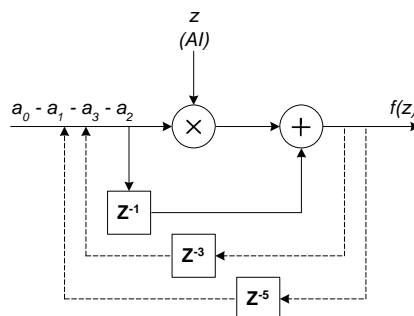


Figure 3: Final reconstruction step (1-D encoding)

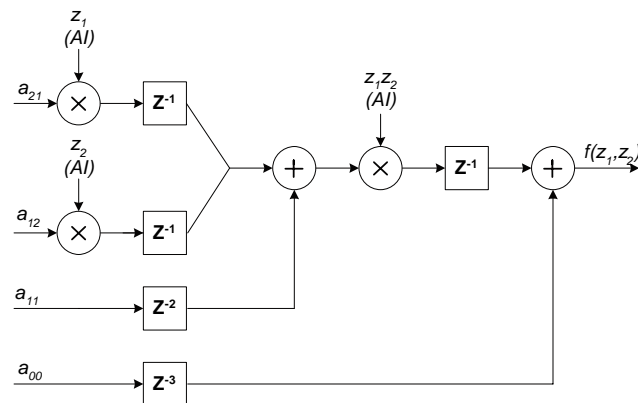


Figure 4: Final reconstruction step (2-D encoding)

For the scaled IDCT, the Linzer and Feig algorithm uses the same fixed multiplier coefficients given by eqn. (3). So, in this case, we can use the same encoding with the same precision described in the above section.

## Comparisons

In Table 5, we compare the computational complexity of previously published AI-based DCT encoding with the proposed scheme. In all cases, the new 2-D AI encoding scheme has the least number of computations.

In Table 6, we present a comparison between some other published 2-D DCT architectures and the proposed algebraic integer approach. Taking the additions as the main computational block, the new multidimensional algebraic-integer-quantization based architecture clearly has the lowest hardware count, particularly considering that all the AI computations are performed without any error.

Table 5: Hardware complexity for different AIQ schemes

Algorithm	Degree of Polynomial	Additions	Shifts	Total Additions
1-D AI-based Chen DCT [Dimitrov, 1998]	7	6	9	156
2-D AI-based Chen DCT [Dimitrov, 2003]	3	3	4	132
AI-based Arai [Wahid, 2004]	3	1	5	30
Proposed 1-D AIQ	3	5	23	32
Proposed 2-D AIQ	2	0	6	29

Table 6: Comparison between different 8-point 2-D DCT

Algorithm	Multiplications	Additions
Linzer-Feig sDCT [Linzer, 1991]	160	416
DCT-SQ [Arai, 1988]	80	464
Distributed DCT [Shams, 2002]	0	672
Proposed 1-D AIQ	0	736
Proposed 2-D AIQ	0	496

---

## Conclusions

---

In this paper, we have introduced a new encoding scheme to compute both 1-D and 2-D Fast Cosine Transform IFCT by error-free mapping of transcendental functions. This new quantization technique effectively reduces the overall number of arithmetic operations, and allows a multiplication-free, parallel, and very fast hardware implementation. Except for the final reconstruction stage, the complete 2-D DCT and IDCT can be implemented without error (infinite precision). This idea of using an algebraic integer quantization scheme can be easily generalized to other algorithms when it is necessary to use real algebraic numbers of special form. Our future work is directed towards the VLSI implementation of this approach for 2-D DCT and IDCT.

---

## Acknowledgements

---

The authors also wish to thank the Alberta Informatics Circle of Research Excellence (iCORE), and the Natural Sciences and Engineering Council (NSERC) of Canada for their financial support for this research. The authors are also indebted to the Canadian Microelectronics Corporation (CMC) for providing the hardware and software infrastructure used in the development of this design.

---

## Bibliography

---

- [Arai, 1988] Y. Arai, T. Agui and M. Nakajima, "A Fast DCT-SQ Scheme for Images", Transactions of Institute of Electronics, Information and Communication Engineers, vol. E71, no. 11, pp.1095-1097, 1988.
- [Baghaie, 2001] R. Baghaie and V. Dimitrov, "Systolic Implementation of Real-valued Discrete transforms via Algebraic Integer Quantization", An International Journal on Computers and Mathematics with Applications, vol. 41, pp.1403-1416, 2001.
- [Dimitrov, 1998] V. S. Dimitrov, G. A. Jullien and W. C. Miller, "A New DCT Algorithm Based on Encoding Algebraic Integers", IEEE International Conference on Acoustics, Speech and Signal Processing, pp.1377-1380, 1998.
- [Dimitrov, 2003] V. Dimitrov and G. A. Jullien, "Multidimensional Algebraic Integer Encoding for High Performance Implementation of the DCT and IDCT", IEE Electronics Letters, vol. 29, no. 7, pp.602-603, 2003.
- [Dedekind, 1996] Richard Dedekind, "Theory of Algebraic Integers", Translated and introduced by John Stillwell, 1996.
- [Duhamel, 1990] P. Duhamel and C. Guillemot, "Polynomial Transform Computation of the 2-D DCT", IEEE International Conference on Acoustics, Speech and Signal Processing, pp.1515-1518, 1990.
- [Feig, 1992] E. Feig and S. Winograd, "Fast Algorithms for the Discrete Cosine Transform", IEEE Transactions on Signal Processing, vol. 40, no. 9, pp.2174-2193, 1992.
- [Knuth, 1981] D. Knuth, "The Art of Computer Programming", vol. 2 - Seminumerical Algorithms, 3rd edition, Addison Wesley, 1981.
- [Linzer, 1991] E. Linzer and E. Feig, "New Scaled DCT Algorithms for Fused Multiply/Add Architectures", IEEE International Conference on Acoustics, Speech and Signal Processing, pp.2201-2204, 1991.
- [Shams, 2002] A. Shams, W. Pan, A. Chidanandan and M. Bayoumi, "A Low Power High Performance Distributed DCT Architecture", Proceedings of IEEE Annual Symposium on VLSI, pp.21-27, 2002.
- [Wahid, 2003] K. Wahid, V. Dimitrov, G. Jullien and W. Badawy, "Error-Free Computation of Daubechies Wavelets for Image Compression Applications", IEE Electronics Letters, vol. 39, no. 5, pp.428-429, March 2003.
- [Wahid, 2004] Vassil Dimitrov, Khan Wahid and Graham Jullien, "Multiplication-Free 8x8 2D DCT Architecture using Algebraic Integer Encoding", IEE Electronics Letters, vol. 40, no. 20, pp.1310-1311, 2004.

---

## Authors' Information

---

Vassil Dimitrov – Dept. of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta, Canada; e-mail: [dimitrov@atips.ca](mailto:dimitrov@atips.ca)

Khan Wahid – Dept. of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta, Canada; e-mail: [wahid@atips.ca](mailto:wahid@atips.ca)

## OPTIMIZATION OF ATM TELECOMMUNICATION NETWORKS

Leonid Hulianytskyi, Andrii Baklan

*Abstract:* ATM network optimization problems defined as combinatorial optimization problems are considered. Several approximate algorithms for solving such problems are developed. Results of their comparison by experiments on a set of problems with random input data are presented.

*Keywords:* network, ATM, optimization, combinatorial optimization, local search, simulated annealing, genetic algorithm

*ACM Classification Keywords:* G.2.1 Combinatorics: Combinatorial algorithms

---

### About Network Optimization Problems

---

Telecommunication and information technologies play a fundamental role in the development of society and economics nowadays. Requirements for telecommunication networks are constantly increasing and, therefore, Broadband Integrated Services Digital Network (B-ISDN) conception has appeared. Evolution of this conception has resulted in appearing of new network technology, universality of which makes it extremely attractive, – Asynchronous Transfer Mode or ATM for short.

The appearance and expansion of this technology along with its high potential make the development of methods for solving ATM network optimization problems important. One of the most important problems is a problem of optimal choice of bandwidth of transmission links for different kinds of traffic for which no effective methods of solution have been developed to present day.

ATM technology is a high-speed, broadband transmission data communication technology based on packet switching (ATM packets are also called cells) and multiplexing technologies and used to carry integrated heterogeneous information, such as data, voice, and video information.

The main requirement for telecommunication networks is the requirement for increasing in their bandwidth together with decreasing in their cost. Availability of high-performance and relatively inexpensive personal computers, workstations, commercial software, expansion of distributed computing – these all demand higher bandwidth at lower cost, available on both local and metropolitan networks. Thus a challenge of providing easy-to-manage broadband services on demand and at an affordable price is arisen.

Different classes of service are used to accommodate transmission of different traffic types in optimal ways, and ATM optimizes traffic flow performance through these various classes of service, which can be allocated on a per-connection basis by using ATM Quality of Service (QoS) settings.

The basic Quality of Service (QoS) parameters (or traffic parameters) that can be negotiated on an ATM network include the following:

- Cell Transfer Delay (CTD);
- Cell Delay Variation (CDV);
- Cell Loss Ratio (CLR);
- Maximum Burst Size (MBS);
- Peak Cell Rate (PCR);
- Sustainable Cell Rate (SCR).

ATM supports several different service categories (kinds of traffic):

- Constant Bit Rate (CBR);
- Variable Bit Rate (VBR);
- Available Bit Rate (ABR);
- Unspecified Bit Rate (UBR).

All these service categories were introduced to attain the ability to transfer heterogeneous traffic, adequate network resources dispatching for each traffic component, more network flexibility and usability. The introducing of service categories increases the advantages making ATM suitable practically for an unlimited range of applications. ATM service categories make it possible for users to choose specific combinations of traffic and performance parameters.

When designing new or analyzing existent telecommunication networks the following problems arise:

- a problem of optimal choice of bandwidth of existent transmission links;
- a problem of optimal choice of routes for transmission and optimal flows distribution;
- a combined problem of optimal choice of bandwidth of existent transmission links, optimal choice of routes for transmission and optimal flows distribution;
- a problem of analysis of survivability indices;
- a problem of network structure synthesis.

One of the most important problems is a problem of optimal choice of bandwidth of transmission links, whose description can be done in the following way. A network structure consisted of switches linked by transmission links is defined. Each transmission link is associated with its length. For each ordered pair of switches the traffic volumes that must be transferred over the network are given. Moreover, an aggregate data flow is given for each transmission link. The bandwidth of each transmission link is proportional to the bandwidth of the basic transmission link. Specific costs per unit length for different bandwidth transmission links are also given. It is necessary to choose such a number of the basic transmission links allocated for traffic for each transmission link that the cost of the network would be minimal and the QoS constraints would be met.

When finding a solution to the problem of optimal choice of bandwidth of transmission links for different kinds of traffic, we must take into account both different kinds of traffic and different QoS parameters for them. There are four different kinds of traffic in ATM technology: CBR, VBR, ABR and UBR, for each of them there are certain QoS parameters negotiated. Having regard to very strong requirement for QoS parameters for CBR traffic, constant bandwidth is allocated for this kind of traffic in each transmission link. VBR and ABR traffics have common bandwidth, which distributes among them in the following way: VBR traffic occupies the greater part of bandwidth and are served by switches with higher relative priority by FIFO procedure, and if there are no VBR cells standing in a queue of a switch, ABR cells are transferred. Finally, the rest of the bandwidth is used for UBR traffic transmission, QoS parameters being not negotiated [1].

Taking into account the above-mentioned, problem description for ATM networks can be formulated in the following way. An ATM network structure consisted of switches linked by transmission links is defined. Each transmission link is associated with its length. For each ordered pair of switches CBR, VBR and ABR traffic volumes that must be transferred over the network are given. Moreover, an aggregate data flow is given for each transmission link. The bandwidth of each transmission link is proportional to the bandwidth of the basic transmission link. Specific costs per unit length for different bandwidth transmission links are also given. It is necessary to choose such numbers of the basic transmission links allocated for CBR traffic, common VBR and ABR traffic, and the share of common VBR and ABR traffic allocated for VBR traffic for each transmission link that the cost of the network would be minimal and the following QoS constraints would be met: CLR and CTD for CBR traffic, CLR and CTD for VBR traffic, CTD for ABR traffic.

The problem of optimal choice of bandwidth of transmission links of ATM network is a new combinatorial optimization problem, to which no effective methods of solution have been developed. Before now only one approach to the problem is proposed – the method of successive analysis and screening of candidate solutions [2]. But this approach don't allow improving solutions found (finding several solutions) and also has essential computational complexities when there is an increase in problem size. Therefore, development of approximate algorithms is expedient to solve the problem.

Statement of the problem of optimal choice of bandwidth of transmission links for CBR traffic as well as VBR and ABR traffics is considered in [2]. Modified statement of the problem under consideration in the form of a combinatorial optimization model is presented below.

---

### Statement of Problem

---

As it is mentioned above, constant bandwidth is allocated for CBR traffic in each transmission link, independently of flow distribution. This allows solving the problem of optimal choice of bandwidth of transmission links for CBR traffic independently of VBR and ABR traffics.

Therefore, we will consider two subproblems: optimal choice of bandwidth of transmission links for CBR traffic and optimal choice of bandwidth of transmission links for VBR and ABR traffics.

#### 1) Optimal choice of bandwidth of transmission links for CBR traffic.

Let  $G = (V, E)$  be an undirected graph that represents a network structure where  $V = \{v_1, v_2, \dots, v_n\}$  is a set of vertices (switches),  $E = \{e_1, e_2, \dots, e_m\}$  is a set of edges (transmission links). Each transmission link  $e_k$  is associated with its length  $l_k$ ,  $k = \overline{1, m}$ . A CBR traffic matrix  $H^{(0)} = \|h_{ij}^{(0)}\|_{i,j=1,n}$  is also given, where  $h_{ij}^{(0)}$  is the traffic volume (Mbit/s) that must be transferred from a switch  $i$  to a switch  $j$ . Moreover, an aggregate CBR traffic data flow vector  $f^{(0)} = (f_1^{(0)} \quad f_2^{(0)} \quad \dots \quad f_m^{(0)})^T$  is given, where  $f_k^{(0)}$  is an aggregate CBR traffic data flow in a transmission link  $e_k$ ,  $k = \overline{1, m}$ . The bandwidth of each transmission link  $e_k$  is proportional to the bandwidth of the basic transmission link  $\mu$  (Mbit/s):  $\mu_k^{(0)} = x_k^{(0)} \mu$ , where  $x_k^{(0)} \in \{1, 2, \dots, N\}$  is the number of the basic transmission links,  $k = \overline{1, m}$ .

Specific costs per unit length for different bandwidth transmission links  $C = \{c_1, c_2, \dots, c_N\}$  are also given. That is the cost of a transmission link  $e_k$  with the number of the basic transmission links  $x_k^{(0)}$  and length  $l_k$  equals to  $c_{x_k^{(0)}} l_k$ ,  $k = \overline{1, m}$ .

It is necessary to choose such numbers of the basic transmission links allocated for CBR traffic  $x^{(0)} = (x_1^{(0)} \quad x_2^{(0)} \quad \dots \quad x_m^{(0)})^T$  that the cost of the network would be minimal:

$$\sum_{k=1}^m c_{x_k^{(0)}} l_k \rightarrow \min \quad (1)$$

provided that the following constraints would be met:

$$CLR^{(0)} \leq CLR_{set}^{(0)}, \quad (2)$$

$$CTD^{(0)} \leq CTD_{set}^{(0)}, \quad (3)$$

$$f_k^{(0)} < x_k^{(0)} \mu, \quad k = \overline{1, m}, \quad (4)$$

$$x_k^{(0)} \in \{1, 2, \dots, N\}, \quad k = \overline{1, m}, \quad (5)$$

where  $CLR^{(0)}$  is the mean probability of losses (the loss ratio) for CBR cells;  $CTD^{(0)}$  is the mean transfer delay for CBR cells;  $CLR_{set}^{(0)}$  is the specified loss ratio for CBR cells;  $CTD_{set}^{(0)}$  is the specified mean transfer delay for CBR cells.

Formulas for  $CTD^{(0)}$  and  $CLR^{(0)}$  obtained in [2] in our notation are the following:

$$CTD^{(0)} = \frac{1}{H_{\Sigma}^{(0)}} \sum_{k=1}^m \frac{f_k^{(0)}}{x_k^{(0)} \mu - f_k^{(0)}}, \quad (6)$$

$$CLR^{(0)} = \frac{1}{m} \sum_{k=1}^m CLR_k^{(0)}, \quad (7)$$

$$CLR_k^{(0)} = P_0 \left( \frac{f_k^{(0)}}{\mu} \right)^{x_k^{(0)}} \frac{1}{x_k^{(0)}!} \left( \frac{f_k^{(0)}}{x_k^{(0)} \mu} \right)^{N_k^{(0)}}, \quad (8)$$

$$P_0 = \left[ \sum_{t=0}^{x_k^{(0)}} \left( \frac{f_k^{(0)}}{\mu} \right)^t \frac{1}{t!} + \left( \frac{f_k^{(0)}}{\mu} \right)^{x_k^{(0)}} \frac{1}{x_k^{(0)}!} \sum_{t=1}^{N_k^{(0)}} \left( \frac{f_k^{(0)}}{x_k^{(0)} \mu} \right)^t \right]^{-1}, \quad (9)$$

where  $H_{\Sigma}^{(0)} = \sum_{i=1}^n \sum_{j=1}^n h_{ij}^{(0)}$  is the aggregate CBR traffic volume;  $N_k^{(0)}$  is the size of a buffer of ATM switch for CBR traffic cells.

## 2) Optimal choice of bandwidth of transmission links for VBR and ABR traffics.

Let  $G = (V, E)$  be an undirected graph that represents a network structure where  $V = \{v_1, v_2, \dots, v_n\}$  is a set of vertices (switches),  $E = \{e_1, e_2, \dots, e_m\}$  is a set of edges (transmission links). Each transmission link  $e_k$  is associated with its length  $l_k$ ,  $k = \overline{1, m}$ . Both a VBR traffic matrix  $H^{(1)} = \|h_{ij}^{(1)}\|_{i,j=1,n}$  and an ABR traffic matrix  $H^{(2)} = \|h_{ij}^{(2)}\|_{i,j=1,n}$  are also given, where  $h_{ij}^{(1)}$  and  $h_{ij}^{(2)}$  are respectively the VBR and ABR traffic volumes (Mbit/s) that must be transferred from a switch  $i$  to a switch  $j$ . Moreover, both an aggregate VBR traffic data flow vector  $f^{(1)} = (f_1^{(1)} \ f_2^{(1)} \ \dots \ f_m^{(1)})^T$  and an aggregate ABR traffic data flow vector  $f^{(2)} = (f_1^{(2)} \ f_2^{(2)} \ \dots \ f_m^{(2)})^T$  are given, where  $f_k^{(1)}$  и  $f_k^{(2)}$  are respectively an aggregate VBR and an aggregate ABR traffic data flows in a transmission link  $e_k$ ,  $k = \overline{1, m}$ . The bandwidth of each transmission link  $e_k$  is proportional to the bandwidth of the basic transmission link  $\mu$  (Mbit/s):  $\mu_k = x_k \mu$ , where  $x_k \in \{1, 2, \dots, N\}$  is the number of the basic transmission links,  $k = \overline{1, m}$ .

Specific costs per unit length for different bandwidth transmission links  $C = \{c_1, c_2, \dots, c_N\}$  are also given. That is the cost of a transmission link  $e_k$  with the number of the basic transmission links  $x_k$  and length  $l_k$  equals to  $c_{x_k} l_k$ ,  $k = \overline{1, m}$ .

It is necessary to choose such numbers of the basic transmission links allocated for common VBR and ABR traffic  $x = (x_1 \ x_2 \ \dots \ x_m)^T$ , and the share of common VBR and ABR traffic allocated for VBR traffic  $x^{(1)} = (x_1^{(1)} \ x_2^{(1)} \ \dots \ x_m^{(1)})^T$  that the cost of the network would be minimal:

$$\sum_{k=1}^m c_{x_k} l_k \rightarrow \min \quad (10)$$

provided that the following constraints would be met:

$$CLR^{(1)} \leq CLR_{set}^{(1)}, \quad (11)$$

$$CTD^{(1)} \leq CTD_{set}^{(1)}, \quad (12)$$

$$CTD^{(2)} \leq CTD_{set}^{(2)}, \quad (13)$$

$$x_k^{(1)} < x_k, \quad k = \overline{1, m}, \quad (14)$$

$$f_k^{(1)} + f_k^{(2)} < x_k \mu, \quad k = \overline{1, m}, \quad (15)$$

$$f_k^{(1)} < x_k^{(1)} \mu, \quad k = \overline{1, m}, \quad (16)$$

$$x_k \in \{1, 2, \dots, N\}, \quad k = \overline{1, m}, \quad (17)$$

$$x_k^{(1)} \in \{1, 2, \dots, N\}, \quad k = \overline{1, m}, \quad (18)$$

where  $CLR^{(1)}$  is the mean probability of losses (the loss ratio) for VBR cells;  $CTD^{(1)}$  and  $CTD^{(2)}$  are the mean transfer delays for VBR and CBR cells respectively;  $CLR_{set}^{(1)}$  is the specified mean probability of losses (the loss ratio) for VBR cells;  $CTD_{set}^{(1)}$  and  $CTD_{set}^{(2)}$  are the specified mean transfer delays for VBR and CBR cells respectively.

Formulas for  $CTD^{(1)}$ ,  $CTD^{(2)}$  and  $CLR^{(1)}$  obtained in [2] in our notation are the following:

$$CTD^{(1)} = \frac{1}{H_{\Sigma}^{(1)}} \sum_{k=1}^m \frac{f_k^{(1)} (f_k^{(1)} + f_k^{(2)})}{x_k \mu (x_k \mu - f_k^{(1)})}, \quad (19)$$

$$CTD^{(2)} = \frac{1}{H_{\Sigma}^{(2)}} \sum_{k=1}^m \frac{f_k^{(2)} (f_k^{(1)} + f_k^{(2)})}{(x_k \mu - f_k^{(1)}) (x_k \mu - f_k^{(1)} - f_k^{(2)})}, \quad (20)$$

$$CLR^{(1)} = \frac{1}{m} \sum_{k=1}^m CLR_k^{(1)}, \quad (21)$$

$$CLR_k^{(1)} = P_0 \left( \frac{f_k^{(1)}}{\mu} \right)^{x_k^{(1)}} \frac{1}{x_k^{(1)}!} \left( \frac{f_k^{(1)}}{x_k^{(1)} \mu} \right)^{N_k^{(1)}}, \quad (22)$$

$$P_0 = \left[ \sum_{t=0}^{x_k^{(1)}} \left( \frac{f_k^{(1)}}{\mu} \right)^t \frac{1}{t!} + \left( \frac{f_k^{(1)}}{\mu} \right)^{x_k^{(1)}} \frac{1}{x_k^{(1)}!} \sum_{t=1}^{N_k^{(1)}} \left( \frac{f_k^{(1)}}{x_k^{(1)} \mu} \right)^t \right]^{-1}, \quad (23)$$

where  $H_{\Sigma}^{(1)} = \sum_{i=1}^n \sum_{j=1}^n h_{ij}^{(1)}$  and  $H_{\Sigma}^{(2)} = \sum_{i=1}^n \sum_{j=1}^n h_{ij}^{(2)}$  are respectively the aggregate VBR traffic volume and the aggregate ABR traffic volume;  $N_k^{(1)}$  is the size of a buffer of ATM switch for VBR traffic cells.



## Experimental Results

For the purpose of making experimental investigation of the developed algorithms a program system implementing local search algorithm (LS) [3], iterated local search algorithm (ILS) [4], simulated annealing algorithm (SA) [5], G-algorithm [6], and genetic algorithm (GA) [7] was developed. This program system also allows generating test problems with random input data.

At the first stage the following parameter values of the algorithms were empirically specified. For ILS algorithm: maximum number of iterations  $t_{\max}=30$ , maximum number of transitions  $h_{\max}=20$ . For SA algorithm: maximum number of iterations  $t_{\max}=20$ ; maximum number of transitions  $h_{\max}=10000$ ; initial temperature value  $T_0=50$ ; temperature schedule coefficient  $r=0,925$ ; number for equilibrium condition determination  $\varepsilon=0,01$ ; number of passages  $k=2$ ; number of transitions per passage  $\nu=35$ . For G-algorithm: maximum number of iterations  $t_{\max}=40$ ; number for equilibrium condition determination  $\varepsilon=10$ ; number of passages  $k=2$ ; number of transitions per passage  $\nu=35$ ; initial value of parameter  $\mu_0=0$ ; parameter  $\gamma=0,05$ . For GA: maximum number of generations  $t_{\max}=500$ ; number of individuals in initial population  $K=20$ ; number of selected individuals for crossing over  $Q=10$ ; parental gene inheritance probability  $P_c=0,1$ ; probability of mutative change of gene  $P_m=0,5$ . Let radiuses of vicinities in LS, ILS and SA algorithms equal 1.

At the second stage numerical experiments on the developed algorithms were made. For that a control set of 80 problem instances, consisted of 16 subsets of 5 problem instances of the same size, was generated. Note that the size of a problem is determined by the number of edges  $m$ .

For each problem instance each algorithm was executed 5 times and, as a result, the following values were found: total execution time  $t$  (ms), the best value of objective function  $f_*$  and improvement  $q$  (%), which is expressed by formula  $q = (f_0 - f_*) / f_0 \cdot 100\%$ , where  $f_0$  is the value of objective function for initial candidate solution.

Average results of numerical experiments on two above-mentioned subproblems are presented in Table 1 for the first subproblem (1)–(9) (CBR traffic) and in Table 2 for the second subproblem (10)–(23) (VBR and ABR traffics).

The program system was developed in Object Pascal programming language in IDE Borland Delphi 7.0. Numerical experiments were run on personal computer with the following characteristics: CPU – AMD Athlon XP 1700+, 1,47 GHz; RAM – DDR 333 MHz, 256 MB; operating system – Windows XP Professional.

Analysis of the results of the experiments has shown that for first subproblem instances a minimal improvement has been given by LS algorithm, other ones have found almost the same solutions, G-algorithm being the best. For second subproblem instances the best solutions have been found by ILS algorithm and G-algorithm, those solutions being almost the same.

Table 1. Average results of numerical experiments on the first subproblem (CBR traffic).

m	IL		ILS		SA		G-algorithm		GA	
	q, %	t, ms	q, %	t, ms	q, %	t, ms	q, %	t, ms	q, %	t, ms
15	31,54	212,20	43,29	1540,20	45,80	12498,00	45,08	8860,60	42,62	4917,20
17	28,54	258,20	44,11	2069,20	46,55	13032,60	45,12	9756,00	43,32	5596,00

20	36,48	450,80	46,15	2889,80	50,00	17735,80	48,89	13120,80	46,07	6609,60
21	15,15	220,00	26,45	2665,80	32,23	14354,60	30,62	11011,80	28,82	6786,00
23	40,74	625,00	53,36	3759,20	57,91	19776,40	56,72	16207,20	53,08	7767,40
25	13,69	308,40	24,40	3623,20	32,65	15636,40	31,09	13481,40	27,21	8217,80
27	31,19	709,00	50,16	5542,20	56,99	24939,60	54,60	18220,20	51,78	8123,60
29	54,04	1382,20	65,86	6673,60	72,37	25462,40	70,14	20828,00	65,28	8632,60
30	28,62	817,00	42,61	6303,20	48,94	28242,80	46,93	20451,40	44,65	8908,80
32	6,83	362,60	27,03	6541,40	40,30	17787,60	38,53	18793,00	36,18	10002,60
33	31,65	1138,00	44,58	8101,60	52,77	31260,80	50,77	23273,40	46,78	9770,20
35	13,24	1019,80	27,83	8884,40	36,74	28689,20	34,62	20770,00	33,28	10152,60
37	40,92	1798,20	56,31	10307,00	64,93	34583,60	62,80	28122,20	60,08	10854,00
40	5,77	619,00	19,61	10086,80	30,52	24569,00	27,44	20215,20	26,03	11813,00
45	10,88	1120,00	29,03	14124,20	44,62	36861,00	41,67	29396,20	38,99	12818,60
50	8,62	1091,60	23,04	15654,60	41,36	36672,40	37,69	31221,40	35,81	14803,00

Table 2. Average results of numerical experiments on the second subproblem (VBR and ABR traffics).

m	IL		ILS		SA		G-algorithm		GA	
	q, %	t, ms	q, %	t, ms	q, %	t, ms	q, %	t, ms	q, %	t, ms
15	75,68	412,60	76,02	2251,20	75,88	9151,20	76,04	5874,60	75,93	4099,80
17	68,82	478,80	71,96	2457,80	72,22	11201,80	71,99	7410,60	71,86	4819,00
20	76,78	787,20	80,18	5013,00	80,28	12269,80	80,05	9711,80	79,14	5834,40
21	67,98	799,00	69,37	4474,80	68,82	13934,00	69,17	8910,80	68,44	5407,60
23	80,57	1021,40	82,93	4867,00	82,95	15246,00	83,13	10126,40	80,95	6060,80
25	59,20	983,40	68,92	8231,80	69,92	16930,40	68,90	12808,60	68,37	6291,00
27	76,12	1434,20	76,35	7036,00	75,42	19414,00	76,39	12283,80	73,61	7514,60
29	84,63	1732,40	84,48	6982,00	83,63	21020,20	84,64	15260,00	80,44	7885,20
30	73,95	1974,00	74,31	9041,20	73,31	22744,60	73,66	15125,80	70,96	8107,60
32	68,62	1946,60	75,68	12577,80	75,38	22801,00	75,53	17947,80	71,79	8530,40
33	74,65	2157,20	75,50	13861,60	74,27	25254,40	75,33	17607,40	71,04	8766,60
35	61,85	2303,40	66,22	14619,20	63,83	27962,20	66,10	17360,60	61,60	9077,20
37	81,30	2794,00	81,48	14088,20	79,92	26670,20	81,50	20229,40	75,54	9882,20
40	65,13	3052,20	66,33	17221,00	63,67	29652,60	65,84	20607,60	60,25	10288,60
45	69,07	3943,40	70,36	20485,80	67,34	34727,60	70,31	24819,80	61,97	11516,60
50	67,47	4708,80	72,81	27754,00	70,02	42056,20	72,59	28853,80	62,42	12770,20

---

## Bibliography

---

1. Назаров А.Н., Симонов М.В. АТМ: технология высокоскоростных сетей. М.: ЭКО-ТРЕНДЗ, 1997.
2. Зайченко О.Ю. Оптимізація характеристик мереж з технологією АТМ // Системні дослідження та інформаційні технології. – 2002. – №3. – С.57– 73.
3. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации – К.: Наук. думка, 1988.
4. Lourenço H. R., Martin O., Stützle T. Iterated local search // Handbook of Metaheuristics: International Series in Operations Research & Management Science, vol. 57 (Eds. F. Glover and G. Kochenberger). – Norwell: Kluwer Academic Publishers, MA, 2002. – P. 321-353.
5. Aarts E., Korst J. Simulated annealing and Boltzmann machines. – New York: John Willey and Sons, 1989.
6. Гуляницький Л.Ф. Решение задач комбинаторной оптимизации алгоритмами ускоренного вероятностного моделирования // Компьютерная математика. – Киев: Институт кибернетики им. В.М. Глушкова НАН Украины, 2004. – №1. – С. 64-72.
7. Reeves C.R., Rowe J.E. Genetic Algorithms: Principles and Perspectives. A Guide to GA Theory. – Boston: Kluwer Academic Publishers, 2002.

---

## Authors' Information

---

Leonid F. Hulianytskyi – V.M.Glushkov Institute of Cybernetics of NAS of Ukraine, Glushkova ave. 40, Kyiv, 03680, Ukraine; e-mail: [lh\\_dar@hotmail.com](mailto:lh_dar@hotmail.com)

Andrii M. Baklan – V.M.Glushkov Institute of Cybernetics of NAS of Ukraine, Glushkova ave. 40, Kyiv, 03680, Ukraine; e-mail: [a\\_baklan@yahoo.com](mailto:a_baklan@yahoo.com)

## ANALYZING THE DATA IN OLAP DATA CUBES\*

Galina Bogdanova, Tsvetanka Georgieva

*Abstract:* OLAP applications provide a possibility to data analysis over large collections of historical data in the data warehouses, supporting the decision-making process. This paper presents an application that creates a data cube and demonstrates the effectiveness of the applying the OLAP operations when it necessary to analyze the data and obtain the valuable information from the data. It allows the analysis of factual data that is daily downloads of folklore materials, according to dimensions of interest.

*Keywords:* data cube, online analytical processing, multidimensional expressions

*ACM Classification Keywords:* H.4.2 Information Systems Applications: Types of Systems – Decision support

---

### 1. Introduction

---

Decision-support functions in a data warehouse, such as online analytical processing (OLAP), involve hundreds of complex aggregate queries over large volumes of data. It is not feasible to compute these queries by scanning the data sets each time [9]. The data cubes are structures designed to provide quick access to the data in data warehouses. The cube definition is determined from the requirements, which the users analyzing the data have

---

\* Supported partially by the Bulgarian National Science Fund under Grant MM-1405/2004

and it is based on the choice of the schema representing dimensional model of the data that consists of one fact table and several dimension tables.

Some requirements of decision-support systems (DSS) are considered in [4]. Several common considerations of designing data cubes are examined in [5]. In [11] an OLAP server supporting dimension updates is presented and view maintenance under these updates is proposed. The advantages of the aggregation ranking queries in OLAP data cubes are described in [6]. In the present paper the OLAP operations are applied to analyze the data in a WEB based client/server system containing archival fund with folklore materials of the Folklore Institute at BAS. It represents the data cube creation and the dimension hierarchies. Part of present paper is reported in [2].

The rest of the paper is organized as follows. Section 2 reviews the concepts of the data cubes, the lattice corresponding to a data cube and the OLAP operations. Section 3 presents the data cube creation and applying the OLAP operations by using the language MDX (*Multidimensional Expressions*). Section 4 gives the conclusion of this paper.

## 2. Data Cubes and OLAP Operations

Data cubes are popular in OLAP because they provide an intuitive way for data analysts to navigate various levels of summary information in the database [9]. In a data cube, attributes are categorized into *dimension attributes*, on which grouping may be performed, and *measures*, which are the results of aggregate functions. From a data cube with  $n$  dimension attributes can be obtained  $2^n$  cube views. For example, a data cube with dimensions  $A_1, A_2, A_3$  is shown in figure 1(a) as a lattice structure.

The dimensions often are organized into dimension hierarchies, which can also be represented by a lattice. For example, figure 1(b) shows the lattice for the dimension hierarchies where  $a_{ij}$  is  $j$ -th level in the hierarchy of the dimension  $A_i$ . The top element of each lattice is "all", meaning no grouping by that dimension.

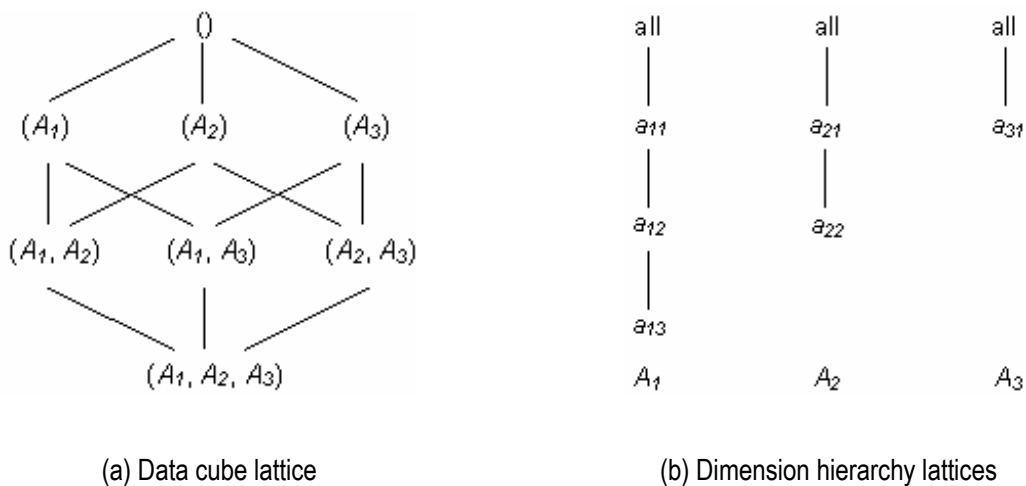


Fig. 1 Example lattices

It can be constructed a lattice representing the set of views that can be obtained by grouping on each combination of elements from the set of the dimension hierarchies. Figure 2 shows the lattice combining the data cube lattice of figure 1(a) with the dimension hierarchy lattices of figure 1(b).

OLAP includes a set of operations for manipulation of the dimensional data organized in multiple levels of abstraction. The basic OLAP operations are roll-up (increasing the level of aggregation); drill-down (decreasing the level of aggregation); slice-and-dice (selection and projection); pivot (re-orienting the multidimensional view of data) [3].

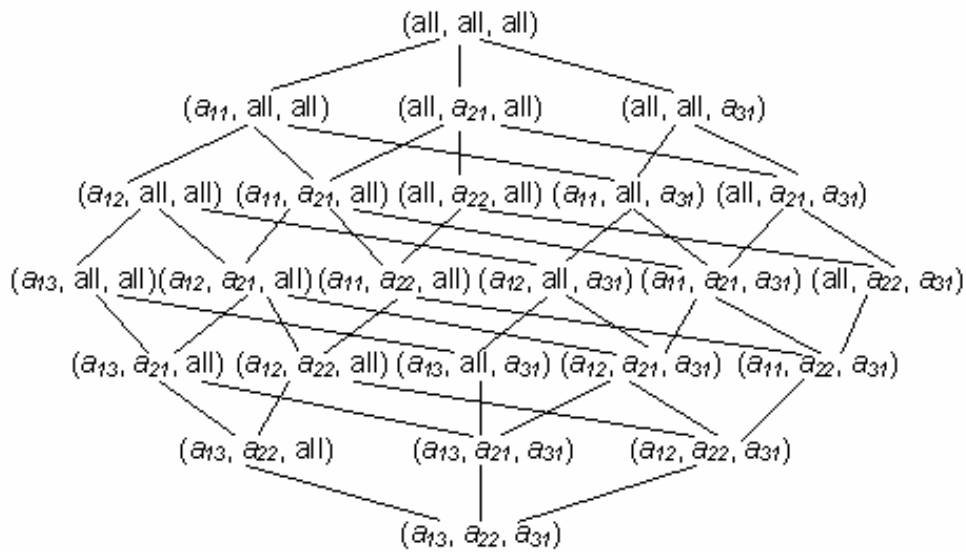


Fig. 2 Combined lattice

### 3. Creation of the Data Cube FolkloreCube and Analyzing the Data by Applying the OLAP Operations

The investigated archive keeps detailed information of the documents and materials, which can be downloaded by the users and contain audio, video and text information.

#### 3.1. The Relational Database FolkloreDB

The OLTP (online transaction processing) database FolkloreDB is created in accordance to the classification schema described in [7]. This database is realized by using the client/server relational database management system (RDBMS) Microsoft SQL Server [8, 10] and consists of the tables shown in figure 3.

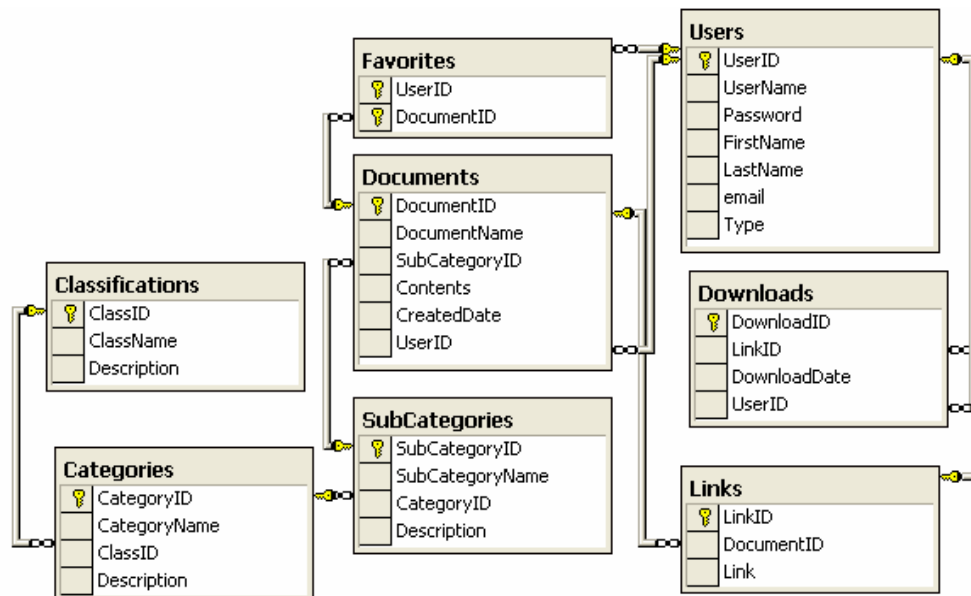


Fig. 3 The structure of the database FolkloreDB

### 3.2. The Data Warehouse Database FolkloreDB\_DW

The database FolkloreDB\_DW in the data warehouse is designed by using the dimensional model represented by the star schema. This database consists of one fact table Downloads\_fact and four dimension tables Documents\_DW, Links\_DW, Users\_DW and Dates (fig. 4).

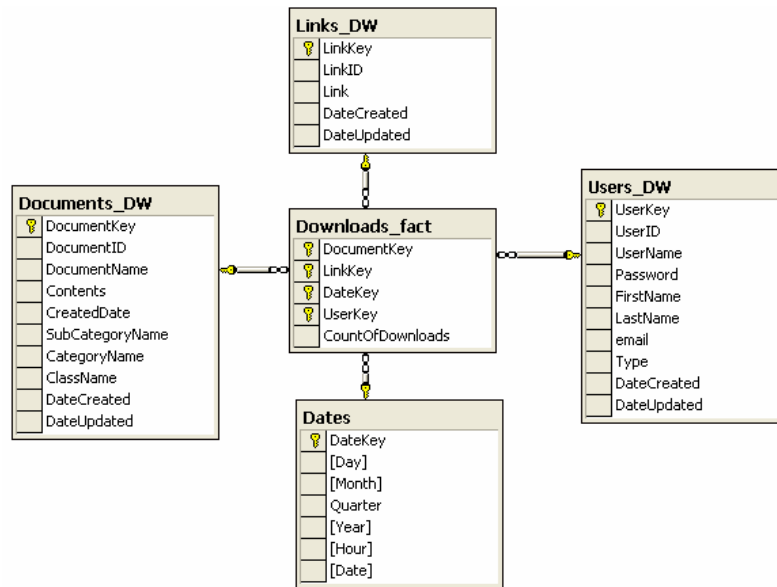


Fig. 4 The structure of the database FolkloreDB\_DW in the data warehouse

### 3.3. Creation of the Data Cube FolkloreCube

The data cube FolkloreCube is created in correspondence with the star schema of the dimensional model of the database FolkloreDB\_DW. The represented statement defines the data cube structure.

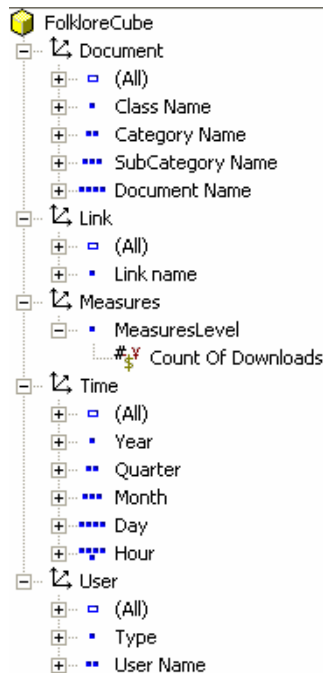


Fig. 5 The structure of the data cube FolkloreCube

```

CREATE CUBE FolkloreCube(
    DIMENSION Document,
        LEVEL [All Documents] TYPE ALL,
        LEVEL [Class name],
        LEVEL [Category name],
        LEVEL [SubCategory name],
        LEVEL [Document name],
    DIMENSION [Link],
        LEVEL [All Links] TYPE ALL,
        LEVEL [Link name],
    DIMENSION [User],
        LEVEL [All Users] TYPE ALL,
        LEVEL [Type],
        LEVEL [User name],
    DIMENSION [Time] TYPE TIME,
        LEVEL [All Time] TYPE ALL,
        LEVEL [Year] TYPE YEAR,
        LEVEL [Quarter] TYPE QUARTER,
        LEVEL [Month] TYPE MONTH,
        LEVEL [Day] TYPE DAY,
        LEVEL [Hour] TYPE HOUR,
    MEASURE [Count of downloads]
        Function Sum)

```

It is executed from the Visual Basic project [1, 12] and creates the local cube (.*cube*) by using Microsoft ActiveX Data Objects and ActiveX Data Objects Multidimensional (ADO MD).

This statement adds data into already defined cube:

```

INSERT INTO FolkloreCube
(Document.[Class name], Document.[Category name],
 Document.[Subcategory name], Document.[Document name],
 Link.[Link name], User.[Type], User.[User name], Time.[Year],
 Time.[Quarter], Time.[Month], Time.[Day], Time.[Hour],
 Measures.[Count of downloads])
SELECT d.ClassName, d.CategoryName, d.SubCategoryName,
       d.DocumentName, l.Link, u.Type, u.UserName, t.Year,
       t.Quarter, t.Month, t.Day, t.Hour, f.CountOfDownloads
FROM Documents_DW d, Downloads_fact f, Links_DW l, Users_DW u,
     Dates t
WHERE d.DocumentKey = f.DocumentKey AND f.LinkKey = l.LinkKey
      AND u.UserKey = f.UserKey AND t.DateKey = f.DateKey

```

The dimension hierarchies of the data cube FolkloreCube are shown in figure 5.

### 3.4. MDX Queries

MDX is Microsoft OLAP query language [13, 14, 15]. MDX queries are applied to the data cube FolkloreCube providing the dimensional view of summarized data.

#### 3.4.1. Top 5 documents from which are downloaded materials (fig. 6);

The execution of the following MDX query provides the result represented in figure 6:

```

SELECT {Measures.[Count of downloads]} ON COLUMNS,
       TOPCOUNT(Document.[Document name].Members, 5,
       Measures.[Count of downloads]) ON ROWS
FROM FolkloreCube

```

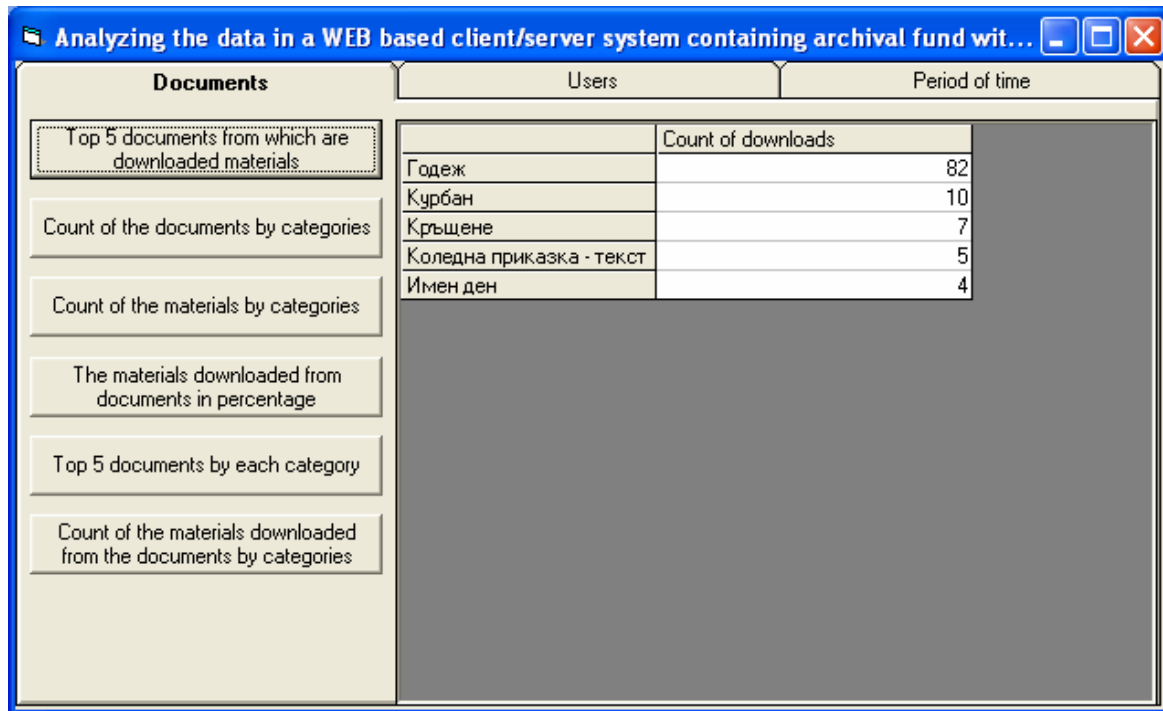


Fig. 6 Exemplary result from applying OLAP operations in FolkloreCube

3.4.2. The documents from which are downloaded the most of materials from the top 10 users (fig. 7);  
 To obtain the result show in figure 7 the represented application performs the MDX query:

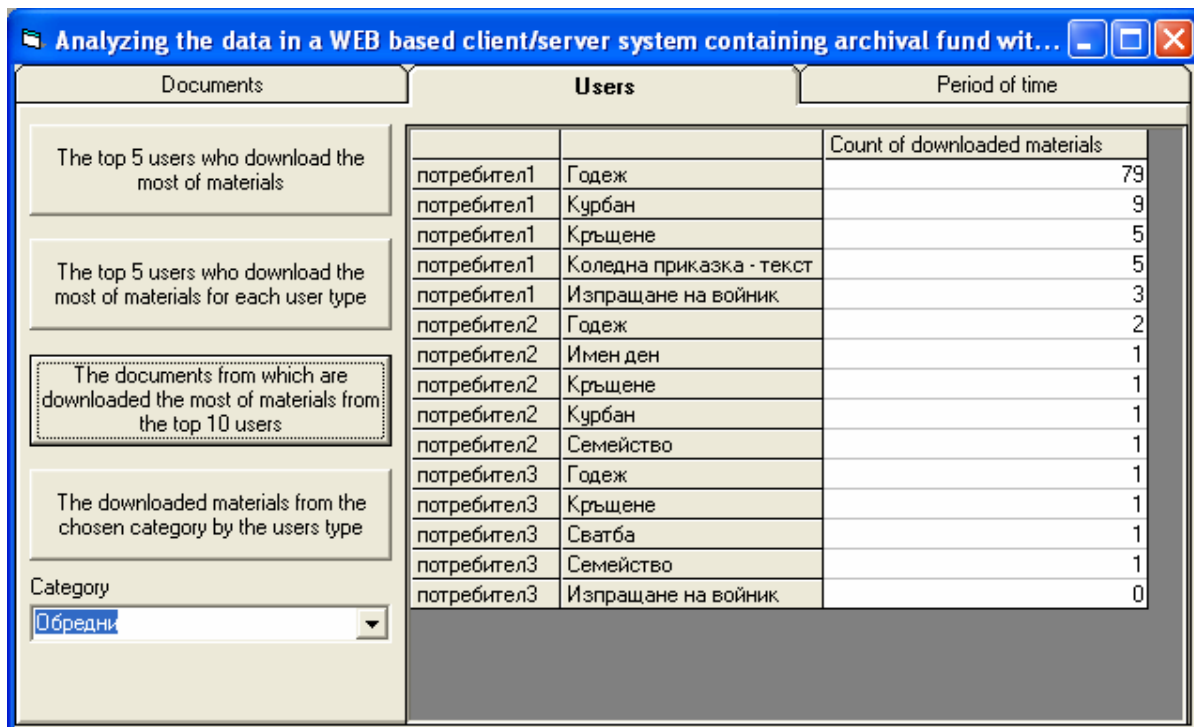


Fig. 7 Example for 3-dimensional view of the data



```

WITH SET Top10Users AS 'TOPCOUNT(User.[User name].Members, 10,
                        Measures.[Count of downloads])'
MEMBER Measures.[Count of downloaded materials] AS
    'CoalesceEmpty(Measures.[Count of downloads], 0)'
SELECT {Measures.[Count of downloaded materials]} ON COLUMNS,
    {GENERATE(Top10Users, CROSSJOIN({User.CURRENTMEMBER},
    TOPCOUNT(Document.[Document name].Members, 5,
    Measures.[Count of downloads])))} ON ROWS
FROM FolkloreCube

```

3.4.3. Count of the materials downloaded from documents by the hours of the chosen date and the difference with the previous hour (fig. 8).

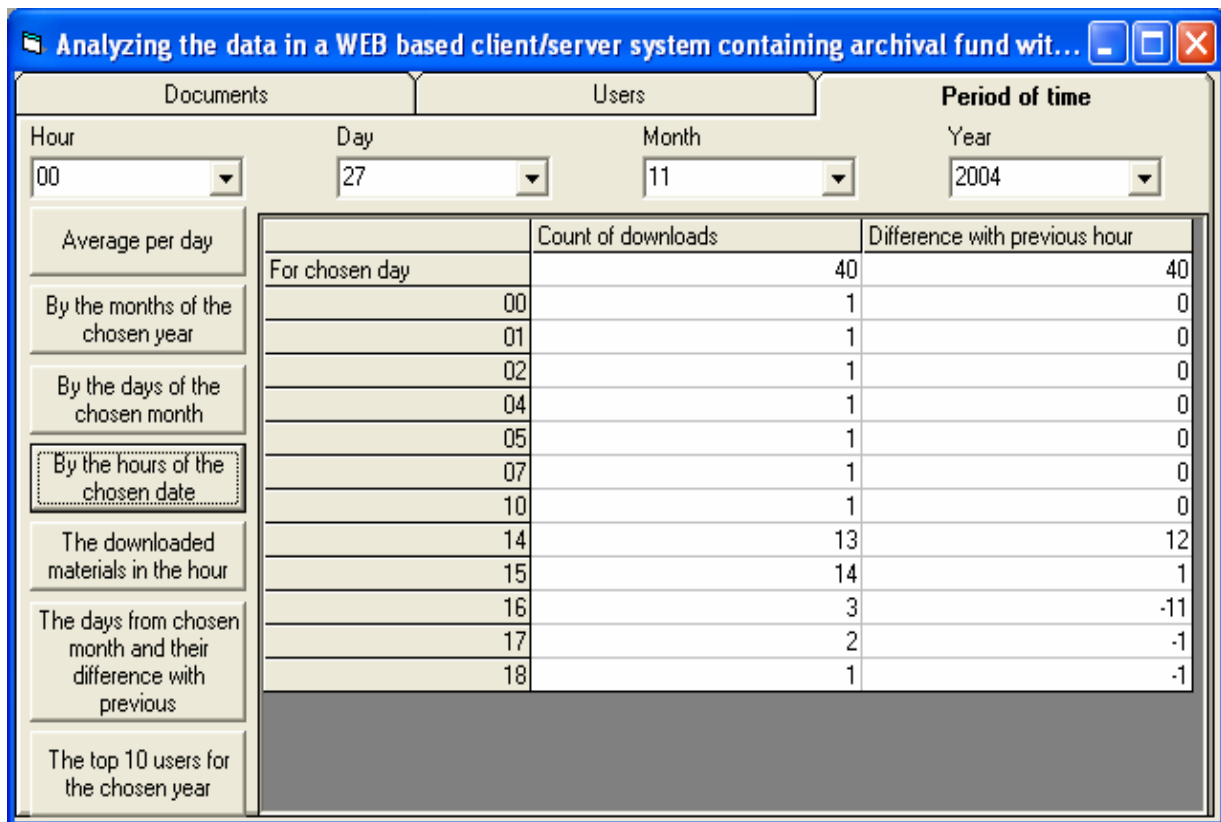


Fig. 8 Analyzing the data in FolkloreCube for a given period of time

```

WITH SET M AS '{Time.[2004].[Q4].[11].[27]}'
MEMBER Time.[For chosen day] AS 'M.Item(0)'
MEMBER Measures.[Difference with previous hour] AS
    '(Measures.[Count of downloads], Time.CURRENTMEMBER) -
    (Measures.[Count of downloads],
    Time.CURRENTMEMBER.PrevMember)'
SELECT {Measures.[Count of downloads],
    Measures.[Difference with previous hour]} ON COLUMNS,
    {Time.[For chosen day],
    GENERATE(M, {Time.CURRENTMEMBER.children})} ON ROWS
FROM FolkloreCube

```

#### 4. Conclusion

---

With rapid developments of data warehouses and OLAP technologies and with enormous amount of data stored in databases in result of daily activities of the organizations it is increasingly important to develop the database applications that convert huge volumes of data into meaningful information. This assists in the decision-making process in different area by providing feedback on past actions of the users and helped to guide future decisions. Represented application gives different views of the data collected in a WEB based client/server system that contains archival fund with folklore materials.

#### Bibliography

---

- [1] Bekuit, B., VB.NET: A Beginner's Guide, AlexSoft, 2002, pages 330.
- [2] Bogdanova, G., Tsv. Georgieva, Applying the OLAP Operations to Analyzing the Data in a WEB based Client/Server System Containing Archival Fund with Folklore Materials, National Workshop on Coding Theory, Bankya, 9-12.12.2004.
- [3] Garcia-Molina, H., J. D. Ullman, J. Widom, Database Systems: The Complete Book, Williams, 2002, pages 1083.
- [4] Georgieva, Tsv., Data Warehousing and OLAP Technology, In Proc. of the International Conference on Computer Systems and Information Technologies, Veliko Tarnovo, 3-5.10.2003, pages 58-65 (in bulgarian).
- [5] Georgieva, Tsv., Designing OLAP Data Cubes, In Proc. of the International Conference on Computer Systems and Information Technologies, Veliko Tarnovo, 3-5.10.2003, pages 106-112 (in bulgarian).
- [6] Li, H., H. Yu, D. Agrawal, A. Abbadi, Ranking Aggregates, Technical Report 2004-07, Department of Computer Science, University of California, Santa Barbara, 2004, pages 14.
- [7] Mateeva, V., I. Stanoeva, Classification Scheme of the Typological Catalogue in the Folklore Institute, Bulgarian Folklore, v. 2-3, 2001, pages 96-109 (in bulgarian).
- [8] Microsoft Corporation, MCSE Training: Microsoft SQL Server 2000 Database Design and Implementation, SoftPress Ltd., 2001, pages 800.
- [9] Mumick, I., D. Quass, B. Mumick, Maintenance of Data Cubes and Summary Tables in a Warehouse, In Proc. ACM SIGMOD Conf. on Management of Data, Tuscon, Arizona, 1997, pages 100-111.
- [10] Soukup, R., K. Delaney, Inside Microsoft SQL Server 7.0, SoftPress Ltd., 2000, pages 952.
- [11] Vaisman, A., A. Mendelzon, W. Ruaro, S. G. Cymerman, Supporting Dimension Updates, In Proc. of the 14<sup>th</sup> International Conference on Advanced Information Systems Engineering, 2002, pages 67-82.
- [12] Wang, W., Visual Basic 6: A Beginner's Guide, AlexSoft, 2002, pages 562.
- [13] <http://www.georgehernandez.com/xDatabases/MD/MDX.htm>
- [14] <http://www.microsoft.com/data/oledb/olap>
- [15] <http://www.microsoft.com/sql>

#### Authors' Information

---

Galina Bogdanova – Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Veliko Tarnovo, P.O.Box: 323, e-mail: [galina@moi.math.bas.bg](mailto:galina@moi.math.bas.bg)

Tsvetanka Georgieva – University of V. Tarnovo "St. St. Cyril and Methodius", Department of Information Technologies; e-mail: [cv.georgieva@uni-vt.bg](mailto:cv.georgieva@uni-vt.bg)

## A MATHEMATICAL APPARATUS FOR DOMAIN ONTOLOGY SIMULATION. LOGICAL RELATIONSHIP SYSTEMS<sup>1</sup>

Alexander Kleshchev, Irene Artemjeva

*Abstract:* A mathematical apparatus for domain ontology simulation will be described in the series of the articles. The goal of this article is to define unenriched and enriched logical relationship systems that can be considered as mathematical models for domain ontologies. The extendable language of applied logic described in the previous articles of the series is used as the language of representation of these systems.

*Keywords:* Extendable language of applied logic, ontology language specification, kernel of extendable language of applied logic, unenriched logical relationship systems, enriched logical relationship systems, enrichment of logical relationship system.

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

---

### Introduction

---

In this article a class of mathematical models called logical relationship systems is defined. For representing these models the extendable language of applied logic described in [Kleshchev et al, 2005 a, 2005b] is used. Unenriched logical relationship systems simulate domain ontologies, their enrichments simulate domain knowledge, and enriched logical relationship systems simulate domains themselves.

---

### 1. An Unenriched Logical Relationship System without Parameters

---

A pair  $O = \langle \Phi, \emptyset \rangle$ , where  $\Phi$  is a semantically correct applied logical theory, having at least one ambiguously interpreted name, will be called an unenriched logical relationship system  $O$  without parameters. The set of propositions for the reduction of  $\Phi$  [Kleshchev et al, 2005 a] will be called the set of logical relationships. All the ambiguously interpreted names of the theory  $\Phi$  will be called unknowns of the system  $O$ . The set of unknowns of the system  $O$  will be designated as  $X$ .

**Example 1.** An unenriched logical relationship system  $O_1 = T_1(\text{ST}, \text{Intervals})$  without parameters representing a simplified model of an ontology for medical diagnostics of acute abdomen diseases.

The logical theory  $T_1(\text{ST}, \text{Intervals}) = \langle \{\text{Definition of partitions}\}, SS_1 \rangle$ , where  $SS_1$  is the following set of propositions.

The sort descriptions for names.

(1.2.1) sort diagnosis: {healthy, pancreatitis}

A diagnosis means the diagnosis of the patient; the diagnosis can be either healthy or pancreatitis.

(1.2.2) sort partition for a sign: {blood pressure, daily diuresis, strain of abdomen muscles}  $\rightarrow$  partitions

A partition for a sign is a function that takes blood pressure, daily diuresis or strain of abdomen muscles and returns a partition of the patient's time axes.

(1.2.3) sort moments of examination: {blood pressure, daily diuresis, strain of abdomen muscles}  $\rightarrow$  ( $\{ \{ [0, \infty) \} \}$ )

---

<sup>1</sup> 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".

Moments of examination means a function that takes blood pressure, daily diuresis or strain of abdomen muscles and returns moments of examining these signs in the patient; time is measured by integer number of hours from the beginning of the patient's examination.

(1.2.4) sort blood pressure: moments of examination(blood pressure)  $\rightarrow$  {normal, high, low}

A blood pressure is a function (process) that takes a moment of the patient's blood pressure examination and returns the value of blood pressure at the moment; the value can be normal, high or low.

(1.2.5) sort strain of abdomen muscles: moments of examination(strain of abdomen muscles)  $\rightarrow$  {absence, presence}

A strain of abdomen muscles is a function (process) that takes a moment of the patient's strain of abdomen muscles examination and returns the value of strain of abdomen muscles at the moment; the value can be absence or presence.

(1.2.6) sort daily diuresis: moments of examination(daily diuresis)  $\rightarrow$  {normal, high, low}

A daily diuresis is a function (process) that takes a moment of the patient's daily diuresis examination and returns the value of daily diuresis at the moment; the value can be normal, high or low.

The unknowns of the system are diagnosis, partition for a sign, moments of examination, blood pressure, strain of abdomen muscles and daily diuresis.

---

## 2. Enriched Logical Relationship Systems Without Parameters

---

If  $O = \langle \Phi_1, \emptyset \rangle$  is an unenriched logical relationship system without parameters and  $\Phi_2$  is such a set of restrictions on the interpretation of names that the logical theory  $\Phi_1 \cup \Phi_2$  is semantically correct [Kleshchev et al, 2005a] then  $S = \langle O, \Phi_2 \rangle$  will be called the enriched logical relationship system without parameters formed from  $O$  by the enrichment  $\Phi_2$ . We will also designate  $S$  as  $\langle \Phi_1, \Phi_2 \rangle$ .

**Example 2.** A possible enrichment for the unenriched logical relationship system  $O_1$  without parameters of example 1.

(2.1) (moment: moments of examination(strain of abdomen muscles)) diagnosis = healthy  $\Rightarrow$  strain of abdomen muscles(moment)  $\in$  {absence}

If the patient is healthy, then at any moment of the patient's strain of abdomen muscles examination its value can be only absence.

(2.2) (moment: moments of examination(blood pressure)) diagnosis = healthy  $\Rightarrow$  blood pressure(moment)  $\in$  {normal}

If the patient is healthy, then at any moment of the patient's blood pressure examination its value can be only normal.

(2.3) (moment: moments of examination(daily diuresis)) diagnosis= healthy  $\Rightarrow$  daily diuresis(moment)  $\in$  {normal}

If the patient is healthy then at any moment of the patient's daily diuresis examination its value can be only normal.

(2.4) diagnosis = pancreatitis  $\Rightarrow$  length(partition for a sign(strain of abdomen muscles)) = 2

If the patient is ill with pancreatitis, then the number of dynamics periods for strain of abdomen muscles is equal to 2.

(2.5) diagnosis = pancreatitis  $\Rightarrow$  length(partition for a sign(blood pressure)) = 2

If the patient is ill with pancreatitis, then the number of dynamics periods for blood pressure is equal to 2.

(2.6) diagnosis = pancreatitis  $\Rightarrow$  length(partition for a sign(daily diuresis)) = 2

If the patient is ill with pancreatitis, then the number of dynamics periods for daily diuresis is equal to 2.

(2.7) (moment: moments of examination(strain of abdomen muscles)  $\cap$  interval(partition for a sign(strain of abdomen muscles), 1)) diagnosis = pancreatitis  $\Rightarrow$  strain of abdomen muscles(moment)  $\in$  {absence}

If the patient is ill with pancreatitis and a moment of examining strain of abdomen muscles belongs to the first dynamics period of the sign, then only value absence can be got.

(2.8) (moment: moments of examination(strain of abdomen muscles)  $\cap$  interval(partition for a sign(strain of abdomen muscles), 2)) diagnosis = pancreatitis  $\Rightarrow$  strain of abdomen muscles(moment)  $\in$  {presence}

If the patient is ill with pancreatitis and a moment of examining strain of abdomen muscles belongs to the second dynamics period of the sign, then only value presence can be got.

(2.9) (moment: moments of examination(blood pressure)  $\cap$  interval(partition for a sign(blood pressure), 1)) diagnosis = pancreatitis  $\Rightarrow$  blood pressure(moment)  $\in$  {normal}

If the patient is ill with pancreatitis and a moment of examining blood pressure belongs to the first dynamics period of the sign, then only value normal can be got.

(2.10) (moment: moments of examination(blood pressure)  $\cap$  interval(partition for a sign(blood pressure), 2)) diagnosis = pancreatitis  $\Rightarrow$  blood pressure(moment)  $\in$  {high}

If the patient is ill with pancreatitis and a moment of examining blood pressure belongs to the second dynamics period of the sign, then only value high can be got.

(2.11) (moment: moments of examination(daily diuresis)  $\cap$  interval(partition for a sign(daily diuresis), 1)) diagnosis = pancreatitis  $\Rightarrow$  daily diuresis(moment)  $\in$  {low}

If the patient is ill with pancreatitis and a moment of examining daily diuresis belongs to the first dynamics period of the sign, then only value low can be got.

(2.12) (moment: moments of examination(daily diuresis)  $\cap$  interval(partition for a sign(daily diuresis), 2)) diagnosis = pancreatitis  $\Rightarrow$  daily diuresis(moment)  $\in$  {normal}

If the patient is ill with pancreatitis and a moment of examining daily diuresis belongs to the second dynamics period of the sign, then only value normal can be got.

(2.13) diagnosis = pancreatitis  $\Rightarrow$  sup(interval(partition for a sign(strain of abdomen muscles), 1)) – inf(interval(partition for a sign(strain of abdomen muscles), 1))  $\in$  R[24, 48]

If the patient is ill with pancreatitis, then the duration of the first dynamics period for strain of abdomen muscles is from 24 to 48 hours.

(2.14) diagnosis = pancreatitis  $\Rightarrow$  sup(interval(partition for a sign(strain of abdomen muscles), 2)) – inf(interval(partition for a sign(strain of abdomen muscles), 2))  $\in$  R[1, 144]

If the patient is ill with pancreatitis, then the duration of the second dynamics period for strain of abdomen muscles is from 1 to 144 hours.

(2.15) diagnosis = pancreatitis  $\Rightarrow$  sup(interval(partition for a sign(blood pressure), 1)) – inf(interval(partition for a sign(blood pressure), 1))  $\in$  R[1, 24]

If the patient is ill with pancreatitis, then the duration of the first dynamics period for blood pressure is from 1 to 24 hours.

(2.16) diagnosis = pancreatitis  $\Rightarrow$  sup(interval(partition for a sign(blood pressure), 2)) – inf(interval(partition for a sign(blood pressure), 2))  $\in$  R[1, 144]

If the patient is ill with pancreatitis, then the duration of the second dynamics period for blood pressure is from 1 to 144 hours.

(2.17) diagnosis = pancreatitis  $\Rightarrow$  sup(interval(partition for a sign(daily diuresis), 1)) – inf(interval(partition for a sign(daily diuresis), 1))  $\in$  R[48, 72]

If the patient is ill with pancreatitis, then the duration of the first dynamics period for daily diuresis is from 48 to 72 hours.

(2.18) diagnosis = pancreatitis  $\Rightarrow$  sup(interval(partition for a sign(daily diuresis), 2)) – inf(interval(partition for a sign(daily diuresis), 2))  $\in$  R[1, 144]

If the patient is ill with pancreatitis, then the duration of the second dynamics period for daily diuresis is from 1 to 144 hours.

If  $S = \langle \Phi_1, \Phi_2 \rangle$  is an enriched logical relationship system without parameters then any model of the applied logical theory [Kleshchev et al, 2005a]  $\Phi_1 \cup \Phi_2$  will be called a solution of  $S$ .

**Example 3.** A solution of the enriched logical relationship system without parameters formed from the unenriched system  $O_1$  of example 1 by the enrichment of example 2 is given by propositions 3.1.10 – 3.1.15 of example 3 of the article [Kleshchev, 2005 b].

---

### 3. An Unenriched Logical Relationship System with Parameters

---

A pair  $O = \langle \Phi, P \rangle$ , where  $\Phi$  is a semantically correct applied logical theory and  $P$  is a nonempty proper subset of the set of ambiguously interpreted names of the theory  $\Phi$ , will be called an unenriched logical relationship system with parameters. The set  $P$  can be given by several ways: by an explicit enumeration of its elements, by a description of a set of names possessing certain properties, by the union of parts of  $P$  given by several ways. The set of propositions for the reduction of the theory  $\Phi$  will be called the set of logical relationships and  $P$  will be called the set of parameters. Ambiguously interpreted names of the theory  $\Phi$  which do not belong to the set  $P$ , will be called unknowns of the system  $O$ . The set of unknowns of the  $O$  will be designated as  $X$ . As it follows from the definition of the set  $P$ , the set  $X$  is not empty.

**Example 4.** The system  $O_2 = \langle T_1(\text{ST, Intervals, Mathematical quantors}), P_2 \rangle$  is an unenriched logical relationship system with parameters where  $T_1(\text{ST, Intervals, Mathematical quantors})$  is the applied logical theory of example 2 [Kleshchev, 2005b] and the set of parameters  $P_2$  consists of the following names: signs, diseases, possible values, normal values, clinical picture, number of dynamics periods, values for a dynamics period, upper bound, lower bound. The unknowns of the system are diagnosis, partition for a sign, moments of examination and also all the names that are elements of a set of names that is an interpretation of parameter signs.

---

### 4. Enriched Logical Relationship Systems with Parameters

---

We will consider two classes of unenriched systems with parameters: pure and mixed.

If  $O = \langle \Phi, P \rangle$  is a pure unenriched logical relationship system with parameters and  $\alpha_P$  is such an interpretation function of the parameters that can be extended to a model of the logical theory  $\Phi$ , then  $S = \langle \Phi, P, \alpha_P \rangle$  will be called an enriched logical relationship system with parameters formed from  $O$  by the enrichment  $\alpha_P$ . The interpretation function  $\alpha_P$  will be called the set of parameter values. An unenriched logical relationship system with parameters will be called pure if its enrichments only of the form  $\alpha_P$  are considered.

**Example 5.** The unenriched system  $O_2$  of example 4 belongs to the class of pure logical relationship systems. The set of parameter values given by propositions 3.1.1 – 3.1.9 of example 3 [Kleshchev, 2005b] is its possible enrichment.

If  $O = \langle \Phi_1, P \rangle$  is a mixed unenriched logical relationship system with parameters and  $\Phi_2$  is such a set of restrictions on the interpretation of names containing no parameters that the logical theory  $\Phi = \Phi_1 \cup \Phi_2$  is semantically correct,  $\alpha_P$  is such an interpretation of the parameters that can be extended to a model of the logical theory  $\Phi$ , then  $S = \langle \Phi, P, \alpha_P \rangle$  will be called an enriched logical relationship system with parameters formed from  $O$  by the enrichment  $\langle \Phi_2, \alpha_P \rangle$ . An unenriched logical relationship system with parameters will be called mixed if its enrichments only of the form  $\langle \Phi_2, \alpha_P \rangle$  are considered. It should be emphasized that the propositions of  $\Phi_2$  contain no parameters. But these propositions can contain unknowns that are constituents of parameter values. We will say that an unknown is a constituent of a parameter value if either the value of the parameter is the unknown or the value of the parameter is a set, tuple or other structure consisting of components and either at

least one of these components is the unknown or the unknown is a constituent of at least one of these components.

Example 6. The mixed unenriched logical relationship system  $O_3 = \langle T_3(\text{ST}, \text{Intervals}), P_3 \rangle$  with parameters representing a model of a simplified ontology of the domain "Masses and volumes of bodies".

The logical theory  $T_3(\text{ST}, \text{Intervals}) = \langle \emptyset, SS_3 \rangle$ , where  $SS_3$  is the following set of propositions.

Value descriptions for names

(6.1.1)  $\text{bodies} \equiv \text{cubes} \cup \text{balls} \cup \text{rectangular parallelepipeds}$

Bodies mean a set of geometric bodies having the form of a cube, a ball or a rectangular parallelepiped.

(6.1.2)  $\text{pi} \equiv 3.1415$

Pi is the well-known mathematical constant.

Sort descriptions for names

(6.2.1) sort cubes:  $\{N\}$

Cubes mean a set of cubes.

(6.2.2) sort balls:  $\{N\}$

Balls mean a set of balls.

(6.2.3) sort rectangular parallelepipeds:  $\{N\}$

Rectangular parallelepipeds mean a set of rectangular parallelepipeds.

(6.2.4) sort radius:  $\text{balls} \rightarrow R(0, \infty)$

A radius is a function that takes a ball and returns the length of its radius.

(6.2.5) sort length of an edge:  $\text{cubes} \rightarrow R(0, \infty)$

A length of an edge is a function that takes a cube and returns the length of its edge.

(6.2.6) sort length:  $\text{rectangular parallelepipeds} \rightarrow R(0, \infty)$

A length is a function that takes a rectangular parallelepiped and returns its length.

(6.2.7) sort width:  $\text{rectangular parallelepipeds} \rightarrow R(0, \infty)$

A width is a function that takes a rectangular parallelepiped and returns its width.

(6.2.8) sort height:  $\text{rectangular parallelepipeds} \rightarrow R(0, \infty)$

A height is a function that takes a rectangular parallelepiped and returns its height.

(6.2.9) sort volume:  $\text{bodies} \rightarrow R(0, \infty)$

A volume is a function that takes a body and returns its volume.

(6.2.10) sort possible substances:  $\{N\}$

Possible substances mean a set of chemical substances.

(6.2.11) sort substance:  $\text{bodies} \rightarrow \text{possible substances}$

A substance is a function that takes a body and returns the chemical substance that the body is made from.

(6.2.12) sort mass:  $\text{bodies} \rightarrow R(0, \infty)$

A mass is a function that takes a body and returns its mass.

(6.2.13) sort density:  $\text{possible substances} \rightarrow R(0, \infty)$

A density is a function that takes a chemical substance and returns its density.

Restrictions on the interpretation of names

(6.3.1)  $(\text{body: bodies}) \text{mass}(\text{body}) = \text{density}(\text{substance}(\text{body})) * \text{volume}(\text{body})$

The proposition represents the well-known relationship among the meanings of terms mass, substance, density, and volume.

The set of parameters  $P_3$  consists of the names possible substances and density. The unknowns are cubes, balls, rectangular parallelepipeds, radius, length of an edge, length, width, height, volume, substance and mass.

A possible enrichment of the system is given by the following propositions

(6.4.1) possible substances  $\equiv \{\text{copper, tin}\}$

Chemical substances copper and tin are only considered.

(6.4.2) density  $\equiv (\lambda (\text{substance: } \{\text{copper, tin}\}) / (\text{substance} = \text{copper} \Rightarrow 8.96) (\text{substance} = \text{tin} \Rightarrow 7.29) /)$

The proposition defines the density of copper and tin.

(6.4.3) (ball: balls) volume (ball)  $= (4 / 3) * (\text{radius}(\text{ball}) \uparrow 3) * \pi$

The proposition defines the well-known formula for calculation of the volume of a ball using its radius.

(6.4.4) (cube: cubes) volume(cube)  $= \text{length of an edge}(\text{cube}) \uparrow 3$

The proposition defines the well-known formula for calculation of the volume of a cube using the length of its edge.

(6.4.5) (rectangular parallelepiped: rectangular parallelepipeds) volume(rectangular parallelepiped)  $= \text{length}(\text{rectangular parallelepiped}) * \text{width}(\text{rectangular parallelepiped}) * \text{height}(\text{rectangular parallelepiped})$

The proposition defines the well-known formula for calculation of the volume of a rectangular parallelepiped using its length, width and height.

Here propositions 6.4.1, 6.4.2 represent the parameter values of the system and propositions 6.4.3 - 6.4.5 represent restrictions on the interpretation of names.

If  $S = \langle \Phi, P, \alpha_P \rangle$  is an enriched logical relationship system, then an interpretation  $\alpha_X$  of unknowns will be called a solution of  $S$  if there is such a model  $\alpha$  of the theory  $\Phi$  that narrowing  $\alpha$  to  $P$  is the same as  $\alpha_P$ , and narrowing  $\alpha$  to  $X$  is the same as  $\alpha_X$ .

**Example 7.** A possible solution of the enriched logical relationship system with parameters of example 5 is the set of unknown values given by propositions 3.1.10 – 3.1.15 of example 3 of article [Kleshchev,2005b].

A possible solution of the enriched logical relationship system with parameters of example 6 can be represented by the following set of value descriptions for names.

(7.1.1) cubes  $\equiv \{\text{ABCDA1B1C1D1}\}$

The only cube is considered.

(7.1.2) rectangular parallelepipeds  $\equiv \emptyset$

The set of rectangular parallelepipeds is empty.

(7.1.3) balls  $\equiv \emptyset$

The set of balls is empty.

(7.1.4) length of an edge  $\equiv (\lambda (\text{cube: } \{\text{ABCDA1B1C1D1}\}) / (\text{cube} = \text{ABCDA1B1C1D1} \Rightarrow 3) /)$

The length of the edge of the cube is equal to 3.

(7.1.5) volume  $\equiv (\lambda (\text{cube: } \{\text{ABCDA1B1C1D1}\}) / (\text{cube} = \text{ABCDA1B1C1D1} \Rightarrow 27) /)$

The volume of the cube is equal to 27.

(7.1.6) substance  $\equiv (\lambda (\text{cube: } \{\text{ABCDA1B1C1D1}\}) / (\text{cube} = \text{ABCDA1B1C1D1} \Rightarrow \text{copper}) /)$

The cube is made from copper.

(7.1.7) mass  $\equiv (\lambda (\text{cube: } \{\text{ABCDA1B1C1D1}\}) / (\text{cube} = \text{ABCDA1B1C1D1} \Rightarrow 241.92) /)$



The mass of the cube is equal to 241.92.

By this means, every enriched logical relationship system determines the set of its solutions. The set of all the solutions for an enriched system  $S$  will be designated as  $A(S)$ . Two enriched logical relationship systems  $S_1$  и  $S_2$  will be called equivalent if  $A(S_1) = A(S_2)$ .

If  $\zeta$  is an enrichment of unenriched system  $O$  then  $S = \langle O, \zeta \rangle$  is  $O$  enriched by  $\zeta$ . An equivalence relation on the set of all possible enrichments of unenriched system  $O$  will be defined by the following way: two enrichments  $\zeta_1$  and  $\zeta_2$  are equivalent if the enriched systems  $\langle O, \zeta_1 \rangle$  and  $\langle O, \zeta_2 \rangle$  are equivalent. The set of the equivalence classes of all the possible enrichments for an unenriched system  $O$  will be designated as  $En(O)$ . If  $k \in En(O)$  is an equivalence class for the set of all possible enrichments of an unenriched system  $O$ , then let  $\langle O, k \rangle = \langle O, \zeta \rangle$ , where  $\zeta \in k$  is an arbitrary representative of the equivalence class  $k$ . In such a manner, an unenriched system  $O$  determines the set of enriched logical relationship systems  $\{\langle O, k \rangle \mid k \in En(O)\}$ .

In what follows, we will consider only such logical relationship systems  $O$  that for all  $k \in En(O)$  every enriched system  $\langle O, k \rangle$  has the following property: for any solution  $\alpha_{x \in A} \in A(\langle O, k \rangle)$  and for any unknown  $x \in X$  the value  $\alpha_x(x)$  contains no ambiguously interpreted names. We will say that a value  $\alpha_x(x)$  contains no ambiguously interpreted names if either  $\alpha_x(x)$  is not an ambiguously interpreted name or if  $\alpha_x(x)$  is a set, a tuple or any other structure consisting of components and none of the components is an ambiguously interpreted name or contains such names. Notice that a parameter value can contain ambiguously interpreted names (both parameters and unknowns).

An unenriched (enriched) logical relationship system will be called predicative if none of its ambiguously interpreted name is a functional name. An unenriched (enriched) logical relationship system will be called functional if none of its ambiguously interpreted name is a predicative name. If unenriched (enriched) logical relationship system is both predicative and functional then it will be called objective. If unenriched (enriched) logical relationship system is neither predicative nor functional then it will be called a system of a general form. The unenriched system of example 7 is a functional logical relationship system.

Special cases of unenriched logical relationship systems have been considered in [Kleshchev et al, 1999], those of enriched logical relationship systems without parameters have been considered in [Artemjeva et al, 1996] and those of systems with parameters have been considered in [Artemjeva et al, 1997].

## 5. Relations among Logical Relationship Systems

To define a relation  $R$  among unenriched logical relationship systems  $O_1, O_2, \dots, O_m$ , an analogous relation  $R'$  among enriched logical relationship systems  $\langle O_1, k_1 \rangle, \langle O_2, k_2 \rangle, \dots, \langle O_m, k_m \rangle$ , where  $k_1 \in En(O_1), k_2 \in En(O_2), \dots, k_m \in En(O_m)$ , and also a relation  $R''$  on the sets  $En(O_1), En(O_2), \dots, En(O_m)$  are introduced. In doing so,  $O_1, O_2, \dots, O_m$  are in the relation  $R$  if and only if for any  $k_1 \in En(O_1), k_2 \in En(O_2), \dots, k_m \in En(O_m)$  from the fact that  $k_1, k_2, \dots, k_m$  are in the relation  $R''$  it follows that the enriched systems  $\langle O_1, k_1 \rangle, \langle O_2, k_2 \rangle, \dots, \langle O_m, k_m \rangle$  are in the relation  $R'$ .

Following the scheme above an equivalence relation between unenriched logical relationship systems will be defined. An unenriched logical relationship system  $O_1$  will be called equivalent to another unenriched logical relationship system  $O_2$  if there is such a one-to-one map  $E$  from the set  $En(O_1)$  onto  $En(O_2)$  that for all  $k \in En(O_1)$  the enriched systems  $\langle O_1, k \rangle$  and  $\langle O_2, E(k) \rangle$  are equivalent [Kleshchev, 2005].

The following statement takes place: for any unenriched logical relationship system with functional parameters there is an equivalent unenriched logical relationship system having no functional parameters. Also the following statement is true: for any unenriched logical relationship system with predicative parameters there is an equivalent unenriched logical relationship system having no predicative parameters.

The following theorem about eliminating parameters of enriched logical relationship systems takes place: for any enriched logical relationship system with parameters and for a given parameter there is an equivalent enriched logical relationship system not containing this given parameter. The proof of the theorem does not differ from the proof of the analogous theorem in [Artemjeva et al, 1997]. The following statement is a corollary of the theorem: for any enriched logical relationship system with parameters there is an equivalent enriched logical relationship system without parameters.

Also the following theorem about eliminating parameters of unenriched logical relationship systems takes place: if  $O = \langle \Phi_1 \cup \Phi_2, P_1 \cup P_2 \rangle$  is an unenriched logical relationship system with parameters where  $\Phi_1$  is the set of the propositions containing no parameters from the set  $P_2$ , then for the mixed unenriched logical relationship system with parameters  $O_1 = \langle \Phi_1, P_1 \rangle$  there is such a completely defined one-valued map  $h$  from the set  $En(O)$  to  $En(O_1)$  that for all  $k \in En(O)$  the enriched system  $\langle O, k \rangle$  is equivalent to the enriched system  $\langle O_1, h(k) \rangle$ . The system  $O_1$  will be called quasiequivalent to the system  $O$ . This theorem is a corollary of the theorem about eliminating parameters of enriched logical relationship systems. It follows from the theorem, in particular, that in general case  $O$  and  $O_1$  are not equivalent because  $\{h(k) | k \in En(O)\} \subset En(O_1)$ . In addition, if the set of parameters  $P_1$  is empty then the quasiequivalent system  $O_1 = \Phi_1$  is an unenriched logical relationship system without parameters. The quasiequivalence relation is reflexive and transitive but antisymmetric.

Now we will define a notion of isomorphism between unenriched logical relationship systems. Two enriched logical relationship systems will be called isomorphic if there is a one-to-one correspondence between the sets of their solutions. An unenriched logical relationship system  $O_1$  will be called isomorphic to an unenriched logical relationship system  $O_2$  if there is such a one-to-one correspondence  $E$  between the sets  $En(O_1)$  and  $En(O_2)$  that for all  $k \in En(O_1)$  the systems  $\langle O_1, k \rangle$  and  $\langle O_2, E(k) \rangle$  are isomorphic.

The following statement takes place: for any unenriched logical relationship system with functional unknowns there is an isomorphic unenriched logical relationship system without any functional unknowns. Also the following statement is true: for any unenriched logical relationship system with predicative unknowns there is an isomorphic unenriched logical relationship system without any predicative unknowns. Moreover, for any functional (predicative) unenriched logical relationship system there is an isomorphic predicative (functional) unenriched logical relationship system.

Further, we will define a notion of homomorphism between unenriched logical relationship systems. An enriched logical relationship system  $S_2$  will be called a homomorphic image of an enriched logical relationship system  $S_1$  if there is a completely defined one-valued map  $h_1$  from the set of solutions of the system  $S_1$  to the set of solution of the system  $S_2$ . In this case we will say that there is a homomorphism  $h_1 : S_1 \rightarrow S_2$ . An unenriched logical relationship system  $O_2$  will be called a homomorphic image of an unenriched logical relationship system  $O_1$  if there is such a completely defined one-valued map  $h$  from the set  $En(O_1)$  to the set  $En(O_2)$  that for all  $k \in En(O_1)$  the system  $\langle O_2, h(k) \rangle$  is a homomorphic image of the system  $\langle O_1, k \rangle$ . In this case we will say that there is a homomorphism  $h : O_1 \rightarrow O_2$ .

Finally, we will define a product of unenriched logical relationship systems. An enriched logical relationship system  $S$  will be called the product of enriched logical relationship systems  $S_1, S_2, \dots, S_m$  (of the product factors) if there are such homomorphisms  $h_1: S \rightarrow S_1, h_2 : S \rightarrow S_2, \dots, h_m : S \rightarrow S_m$  that for any  $\alpha'X, \alpha''X \in A(S)$  the statement  $\alpha'X \neq \alpha''X \Rightarrow \langle h_1(\alpha'X), h_2(\alpha'X), \dots, h_m(\alpha'X) \rangle \neq \langle h_1(\alpha''X), h_2(\alpha''X), \dots, h_m(\alpha''X) \rangle$  is true. An unenriched logical relationship system  $O$  will be called the product of unenriched logical relationship systems  $O_1, O_2, \dots, O_m$  (of the product factors),  $O = O_1 \otimes O_2 \otimes \dots \otimes O_m$ , if there are such homomorphisms  $h_1: O \rightarrow O_1, h_2: O \rightarrow O_2, \dots, h_m: O \rightarrow O_m$  that for any  $k', k'' \in En(O)$  the statement  $k' \neq k'' \Rightarrow \langle h_1(k'), h_2(k'), \dots, h_m(k') \rangle \neq \langle h_1(k''), h_2(k''), \dots, h_m(k'') \rangle$  is true and for all  $k \in En(O)$  the system  $\langle O, k \rangle$  is the product of the systems  $\langle O_1, h_1(k) \rangle, \langle O_2, h_2(k) \rangle, \dots, \langle O_m, h_m(k) \rangle$ .

---

## Conclusions

---

In this article unenriched logical relationship systems are introduced on the basis of the applied logic languages. Every such a system represents a class of enriched logical relationship systems. Every enriched logical relationship system determines a set of its solution. In this article notions of equivalence, isomorphism, homomorphism, and product are introduced for unenriched logical relationship systems.

---

## References

---

- [Artemjeva et al, 1996] Artemjeva I.L., Gavrilova T.L., Kleshchev A.S. Logical Relationship Systems with Elementary Objects. In Scientific and Technical Information, Series 2, 1996, 1: 11-18 (in Russian).
- [Artemjeva et al, 1997] Artemjeva I.L., Gavrilova T.L., Kleshchev A.S. Logical Relationship Systems with Parameters. In Scientific and Technical Information, Series 2, 1997, 7: 19-23 (in Russian).
- [Kleshchev et al, 1999] Kleshchev A.S. & Artemjeva I.L. Domain Ontologies and Knowledge Processing. Techn. Report 7-99, Vladivostok: Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences, 1999. 25 p.
- [Kleshchev et al, 2005a] 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.
- [Kleshchev et al, 2005b] Kleshchev A.S. and Artemjeva I.L. A mathematical apparatus for ontology simulation. Specialized extensions of an extended language of applied logic. In International Journal Information Theories & Applications. 2005.

---

## Authors' Information

---

Alexander S. Kleshchev – [kleshchev@iacp.dvo.ru](mailto:kleshchev@iacp.dvo.ru)

Irene L. Artemjeva – [artemeva@iacp.dvo.ru](mailto:artemeva@iacp.dvo.ru)

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

## GRID-ENABLING SATELLITE IMAGE ARCHIVE PROTOTYPE FOR UA SPACE GRID TESTBED<sup>1</sup>

Andrey Shelestov, Alexander Lobunets, Michael Korbakov

*Abstract: The paper describes practical approach to implementation of satellite data archive using Globus Toolkit 4 components. The solutions consists in converting a hierarchy of remote data files available via FTP into Grid-enabled archive. All etries of such archive will be indexed using arbitrary but pre-defined XML schema. The information will be exposed via MDS4 Index service and the actual data will be exposed via GridFTP. The schema used in our solution is simple enough for understanding but in a real life applications we should use metadata standards such as ISO 19139 and ISO 19115 in particular. A working prototype of the archive described in this paper is deployed on the Grid testbed of Space Research Institute of National Academy of Science and National Space Agency of Ukraine (SRI-NASU-NSAU). The SRI-NASU-NSAU testbed is briefly described in this paper as well.*

*Keywords: Grid, distributed systems, parallel computing, satellite data, image processing, file archive, programming languages.*

*ACM Classification Keywords: C.2.4 Distributed Systems: Distributed applications*

---

<sup>1</sup> The work is partially supported by the grant of National Academy of Science of Ukraine for the support of scientific researches by young scientists № 24-7/05, " Розробка Desktop Grid-системи і оптимізація її продуктивності ".

## Introduction

Grid systems are becoming a standard solutions for enabling remote computations execution and distributed data access and processing in environments of the different level of scalability. As it was stated by Ian Foster "The real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.... This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organization (VO)" [Anatomy, 2001].

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

Space agencies all over the world have started working on evaluation of Grid technology in their application area. Space data processing is an important subject domain for application of Grid computing among high energy physics, biology, economics, etc, which produces huge amounts of data. Thus, problems of storing, indexing for quick retrieval on application's demand arise within mentioned above the area. One of the possible solutions of the problem of organization of Grid-enabled data archive is presented below. The development was performed within the framework of UA Space Grid project.

## UA Space Grid Testbed Environment

Among different Grid research tasks the most challenging one is the establishment of production-use Grid infrastructure or at least working prototype. This activity includes creation and maintenance of both technological (development, deployment, testing, evaluation of software components) and political (covering managerial issues) infrastructures.

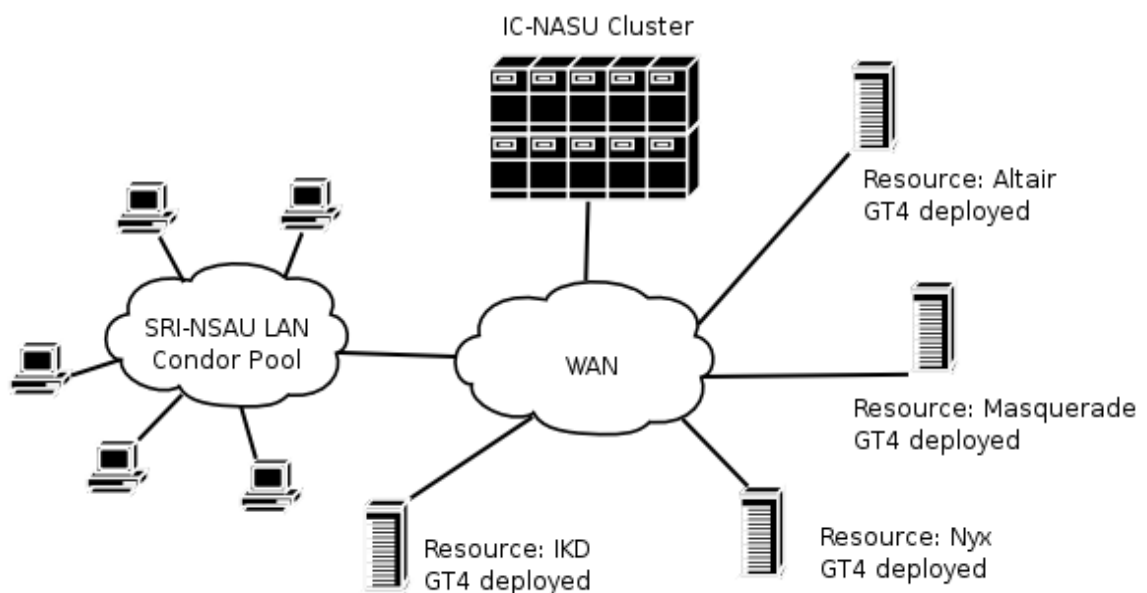


Fig. 1. Current structure of UASG Virtual Organization testbed.

Space Research Institute NASU-NSAU (SRI-NASU-NSAU) is developing its own Grid testbed under the draft name UA Space Grid (UASG). The primary focus of the UASG is to integrate available computing resources

of NASU institutions such as Institute of cybernetics NASU (IC-NASU) for solving spacial data processing related problems [KSPKK, 2005]. These problems include:

- Data preprocessing (noise reduction, quality adjustment);
- Cloud mask extraction (algorithms are based on Markov Random Fields approach);
- Fractal features extraction (modelling based on fractional Brownian Motion process).

The current structure of UASG is depicted in Figure 1.

UASG represents is single Virtual Organization (VO) with a Certification Authority (CA) located in SRI-NASU-NSAU. Its basic computing elements reside in the SRI-NASU-NSAU, IC-NASU and 3<sup>rd</sup> party's co-locations in Kyiv, Ukraine. Currently most of the resources are dynamic. This perplexes the maintenance activities and requires a flexible and reliable monitoring system deployed over the testbed to provide operational information about resources availability.

The testbed is based on Globus Toolkit 4 (GT4) framework [<http://www.globus.org>] which is known as the "de facto standard" by numerous agencies. GT4 provides the following components for implementation of Grid infrastructures:

- Security infrastructure based on X.509 certificates
- Data management services (such as data replication, secure and reliable file transfer)
- Execution management services (WS GRAM, Pre-WS GRAM, WMS, CSF)
- Information services (Index service, Trigger service, Aggregation framework)

In this paper we would like to emphasize the WS GRAM (Web Services based Grid Resource Allocation & Management) component [WS-GRAM, 2005]. It provides a basic mechanism for initiation, monitoring, management, scheduling, and/or coordination of remote computations via Web Services based interfaces. A typical remote execution workflow using WS GRAM is depicted in Figure 2.

According to Figure 2 a user signs in into Grid testbed using his/her X.509 certificate, arranges a job description and send job descriptor to WS GRAM service. A general job descriptor contains the name of the executable to be launched remotely, arguments, logging options, files staging in and out options. A WS GRAM service performs file staging options to retrieve all required data for job execution and contacts Local Resource Scheduler (for example, Linux fork command [<http://www.linuxforum.com/man/fork.2.php>], Condor scheduler [Condor, 2005], etc) which performs the actual execution. Files staging in and out is performed via GridFTP and RFT (Reliable File Transfer) components.

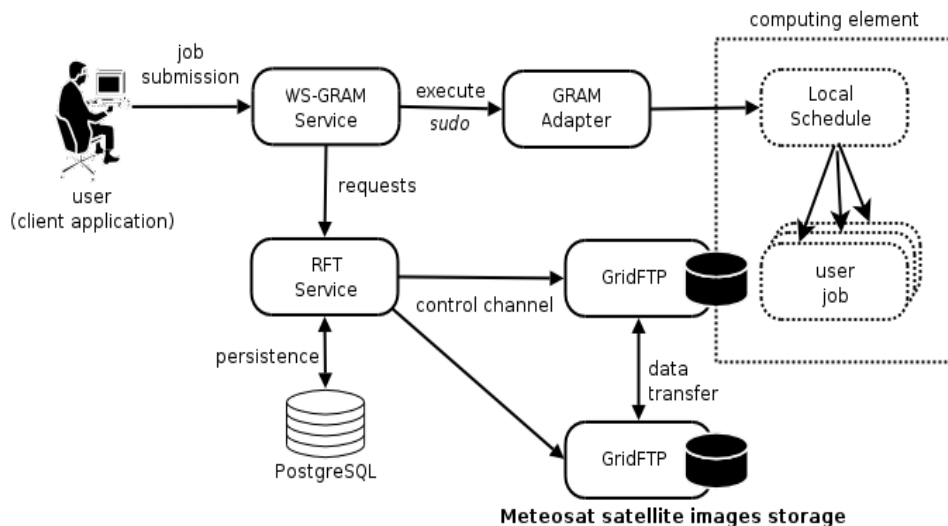


Fig.2. Remote execution with WS GRAM

Another important GT4 component heavily used by application within UASG testbed is Information management which is represented by Monitoring and Discovery System (MDS). MDS "is a suite of web services to monitor and

discover resources and services on Grids. This system allows users to discover what resources are considered part of a *Virtual Organization (VO)* and to monitor those resources. MDS services provide query and subscription interfaces to arbitrarily detailed resource data and a trigger interface that can be configured to take action when pre-configured trouble conditions are met" [MDS,2005]. MDS includes a framework for data acquisition and aggregation, an WSRF [<http://www.globus.org/wsrfl>] compliant Index and Trigger services.

### Sample Applications

For testing and benchmarking purposes we've developed a set of applications that use standard data processing algorithms often found in space science. This application also solves some practical tasks of UASG. This set of applications consists of:

- cloud mask extraction (heterogeneous Markovian segmentation and motion detection algorithms)
- fractal-based analysis of atmosphere (approximation by the fractional Brownian motion)
- visualization (spatiotemporal interpolation, 3d-modeling)

The problem of cloud mask extraction does not put any special requirements for data archive. Depending on underlying algorithm it requires one satellite image or a short sequence of them. The particular algorithms used is heterogeneous Markovian segmentation and optical flow motion analysis (early stage of development). Fractal-based cloud modeling have a similar requirements for data archive. However data visualization is putting much more stronger requirements on underlying data services. Data visualization requires heterogeneous data such as basic topography, DEM data and current satellite images for processing.

Each application follows the same business logic (or workflow) with the only difference in executable file and input/output data and visualization service location. A typical sample application workflow is depicted in Figure 3:

A client daemon component of the sample application runs continually and relies on MDS4 Trigger service for monitoring newly appeared image in the archive. "New image event" triggers a job submission logic which composes an XML job descriptor [WS-GRAM, Int] and submits the job to WS GRAM service. A typical submitted job has the following structure:

- Perform file staging in (Download required input files from specified sources)
- Execute processing applications on the resource (The executable will produce result files)
- Upload/Copy/Move result files to the visualization services location (For instance, a WWW directory of a MapServer)

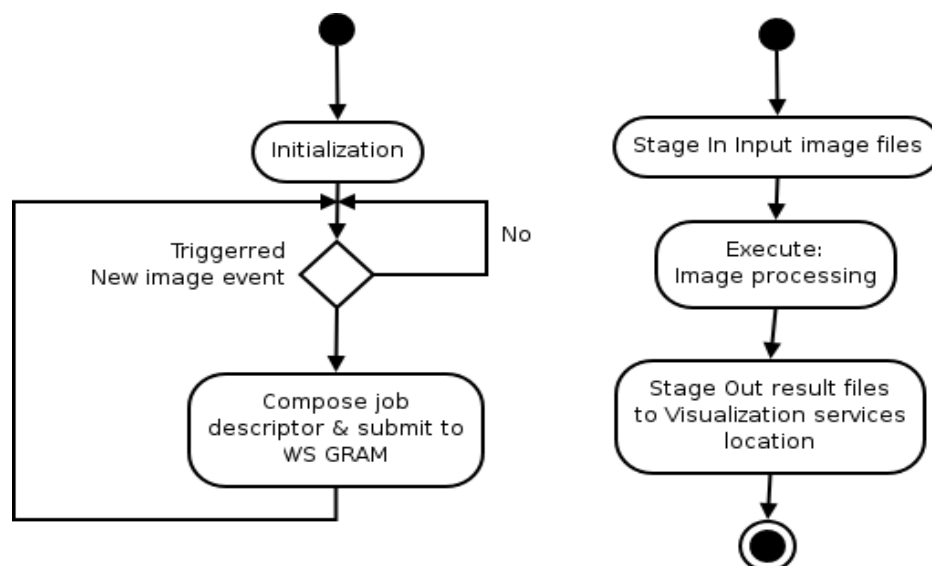


Fig.3. Sample application workflow. Left: client daemon. Right: Job structure.

All applications mentioned above require access to satellite data provided by receiving station. A specific metainformation should be attached to each file which describes image-specific attributes such as range, resolution, dimensions, date, etc. At the moment a very simple scheme is used for metadata with a very limited set of fields such as dimensions, satellite band and time. In future we plan to use standard schemas for metadata, ISO 19139 and ISO 19115 in particular.

Thus a unified interface to data files is required which will be based on Grid concepts and utilize available in the UAGS technologies to avoid formation of so called shop of frameworks or technologies.

The GT4 framework which is used for development of UASG provide all basic services required for setting up a Grid-enabled archive.

### Archive Architecture Overview

As it was mentioned in previous chapter, a unified interface to spacial data is required for application. This will avoid increase of technologies types in the system and simplify developer's efforts in accessing data in the Grid. Unfortunately, no elegendible solution was found in respect to implementation of Grid interface to file archive. As a result, the architecture featured in Figure 4 was proposed to support sample application within UASG.

Let us describe the Grid-enable file archive idea illustrated with Figure 4 in more details.

The system consists of the following components: receiving station, archive host and VO index service. Receiving station is a 3<sup>rd</sup> party organization which periodically uploads satellite image files to archive host filesystem via dedicated line using well known File Transfer Protocol (FTP). This data flow is marked with number "1" in Figure 4.

The archive host consists of the following subcomponents:

- Filesystem
- Execution aggregator source
- MDS4 Index service
- GridFTP server

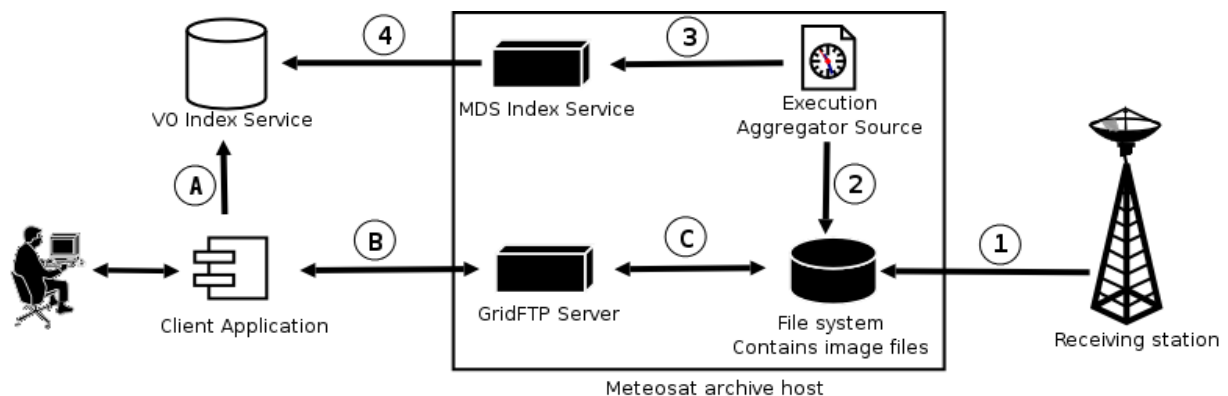


Fig. 4. Satellite data archive architecture

Execution aggregator source is a MDS4 component that performs periodical lookup to filesystem, gathers metainformation and aggregates it into local MDS4 Index service (a subcomponent of archive host as well). Thus, Index service is a temporal storage of metainformation. The persistence of data can be achieved using MDS4 Archive service which is currently under the development. If the Index service will be restarted or will reset its data the metainformation will be re-acquired and in this way database will be recovered.

Metainformation acquisition and aggregation is marked with numbers "2" and "3" accordingly in Figure 4. The aggregator source script was implemented with Ruby programming language [Ruby] using external FTP command line tools such as [NCFTP-Client]. The Ruby language was chosen as a scripting language for sample

applications in UASG because of its simplicity and tight integration with standard UNIX shell. This features make Ruby the perfect alternative to bash or csh scripts that tends to become very obscure with time.

The simplified script is provided in Listing 1 below:

```
#!/usr/bin/ruby
$listCommand = "ncftpls -x \"-1\"""
$server = "ftp://ikd.d23.dedicated"
$rootFolder = "/pub/meteosat"
$urlPrefix = "archive.d23.local:2811/pub/meteosat/"
def getDirectoryInfo(dirName, channelName)
  ...(skipping lines)
end
result = Array.new
result += getDirectoryInfo "LOOP15", "VIS"
puts "<?xml version=\"1.0\" standalone=\"yes\"?>"
puts "<MeteosatInformation>"
result.each do |item|
  puts "\t<MeteosatPhoto>"
  ... (skipping lines)
  puts "\t</MeteosatPhoto>"
end
puts "</MeteosatInformation>"
```

Listing 1. Simplified aggregator script

All metainformation, which is used in the script listed above, is acquired inside *getDirectoryInfo* function's body. Obtained metainformation is converted into Extensible Markup Language (XML) and printed into standard system output. The output is read by MDS aggregation framework, parsed and included into Index service's Resource Properties.

The full version of the script is available at the URL: <http://nyx.linuxside.net:2500/globus/show/HomePage>.

The process or registration of a new execution aggregator source is well documented in [Miller, 2005].

A GridFTP server component serves as provider of data files on user's request by given direct URL.

And, finally, local MDS4 Index service registers itself (in other words, publishes) at centralized Index service of VO or so called Grid Information Service (marked as "4" in Figure 4). Thus information about satellite images becomes available to all VO users.

A typical scenario of interaction with image archive will be reviewed in the next chapter.

---

## Examples of Interaction

---

Consider the typical scenario of Grid user interaction with Grid-enabled image archive. First of all user logs into VO using his/her credentials. For instance:

```
alex@eclipse ~ $ grid-proxy-init
Your identity: /O=Space Research Institute/OU=D23/CN=Alexander Lobunets
Enter GRID pass phrase for this identity: *****
Creating proxy ..... Done
Your proxy is valid until: Sun Sep 25 17:23:19 2005
```

Listing 2. Initialization of working in Grid.

Then a user query VO Index service for available satellite data (in our case - meteosat image files) using common tool *wsrf-query* provided by GT4 for querying Resource Properties of WSRF service:

```
alex@eclipse ~ $ wsrf-query -s \
https://gis.d23.local:8443/wsrf/service/DefaultIndexService \
"//*[local-name()='MeteosatInformation']"
```

Listing 3. Querying VO Index service.



In the listing above a user specified a URL of VO Index service using “-s” option (in current example: <https://gjs.d23.local:8443/wsrf/service/DefaultIndexService> ). The last argument is an expression in XPath language that allows to specialize the exact data to retrieve. The result of query execution is provide in Listing 4.

```
alex@eclipse ~ $
<?xml version="1.0" standalone="yes"?>
<MeteosatInformation>
  ... (skipping lines)
  <MeteosatPhoto>
    <Year>05</Year>
    <Month>07</Month>
    <Day>29</Day>
    <Hour>03</Hour>
    <Minute>51</Minute>
    <Channel>VIS</Channel>

    <URL>archive.d23.local:2811/pub/meteosat/LOOP1505072903.515.tiff</URL>
  </MeteosatPhoto>
  ... (skipping lines)
</MeteosatInformation>
```

Listing 4. Query output.

Mentioned above actions are marked with “A” in Figure 4. As you can see from Listing 4 a user receives detailed information about image file including direct URL to image file via GridFTP server. Next, a user can decide to transfer an image he/she is interested in using common commandline tool *globus-url-copy* for example (marked as “B” in Figure 4):

```
alex@eclipse ~ $ globus-url-copy -vb \
gsiftp://archive.d23.local:2811/pub/meteosat/LOOP1505072903.515.tiff \
file:///home/alex/
```

Listing 5. Transferring selected file via archive GridFTP server.

---

## Conclusions

In this paper there was provided practical approach to implementation of a Grid-enabled satellite data archive. That was only a working prototype description aimed to evaluate concepts of basic services provided by GT4 application. This approach can be applied to any kind of file archive and is presumed as general approach for solving similar issues.

In a production use Grid-systems large amounts of data are distributed among different host which will require more sophisticated solutions for implementing efficient archiving. Besides processing logic will require to replicate subsets of the archives into locations which are efficient from the processing performance point of view. These issues provide reasons for continuing research in this area and finally arranging the general approach to organization and maintenance of satellite data archive with a Grid-enabled access.

---

## Bibliography

- [Anatomy, 2001] I. Foster, C.Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations International Journal of High Performance Computing Applications, 15 (3). 200-222. 2001. Available at <http://www.globus.org/research/papers/anatomy.pdf>
- [KSPKK, 2005] N. Kussul, A. Shelestov, N. Phuong, M. Korbakov, A. Kravchenko. Parallel Markovian Approach to the Problem of Cloud Mask Extraction, Proceedings of XI-th International Conference “Knowledge-Dialog-Solution”, June 20-30, 2005, Varna Bulgaria, pp.567-569.
- [WS-GRAM, 2005] WS-GRAM Documentation // <http://globus.org/toolkit/docs/4.0/execution/wsgram/>

[WS-GRAM, Int] WS GRAM Public Interfaces // Available at

[http://www.globus.org/toolkit/docs/4.0/execution/wsgram/WS\\_GRAM\\_Public\\_Interfaces.html](http://www.globus.org/toolkit/docs/4.0/execution/wsgram/WS_GRAM_Public_Interfaces.html)

[Condor, 2005] Condor Project Home // <http://cs.wisc.edu/condor/>

[MDS, 2005] Information Services (MDS) : Key Concepts // <http://globus.org/toolkit/docs/4.0/info/key-index.html#id2770162>

[Ruby] Ruby Programming Language // <http://www.ruby-lang.org>

[NCFTP-Client] A command line ftp client for Unix // Available at <http://www.ncftp.com/ncftp/>

[Miller, 2005] Nail Miller. How to write a simple Execution Aggregator Information Provider for MDS4 // Available at [http://www-unix.mcs.anl.gov/~neillm/mds/writing\\_a\\_gt4\\_information\\_provider.html](http://www-unix.mcs.anl.gov/~neillm/mds/writing_a_gt4_information_provider.html)

[Raicu, 2005] Ioan Raicu. A performance study of the Globus Toolkit and Grid services via DiPerF, an automated distributed performance testing framework // M.S. Thesis, The University of Chicago, May 2005

---

### Author's Information

---

Andrey Y. Shelestov, PhD. – senior scientific researcher, Space Research Institute NASU-NSAU; 40 Glushkov Ave 03187, Kyiv, Ukraine; e-mail: [inform@ikd.kiev.ua](mailto:inform@ikd.kiev.ua)

Alexander G. Lobunets, MSc. – junior scientific researcher, Space Research Institute NASU-NSAU; 40 Glushkov Ave 03187, Kyiv, Ukraine; e-mail: [alexander.lobunets@gmail.com](mailto:alexander.lobunets@gmail.com)

Michael B. Korbakov, Bcs. – system developer, Space Research Institute NASU-NSAU; 40 Glushkov Ave 03187, Kyiv, Ukraine; e-mail: [alexander.lobunets@gmail.com](mailto:alexander.lobunets@gmail.com)

---

## REPRESENTATION OF NEURAL NETWORKS BY DYNAMICAL SYSTEMS

Volodymyr Donchenko, Denys Serbaev

*Abstract:* Representation of neural networks by dynamical systems is considered. The method of training of neural networks with the help of the theory of optimal control is offered.

*Keywords:* neural nets, dynamical systems, training.

*ACM Classification Keywords:* G.1.6 Optimization, G.1.2 Approximation, I.2 Artificial Intelligence

---

### Introduction

---

At present time neural networks have received the wide circulation and are successfully applied to the decision of various complicated problems such as, for example, control and identification of nonlinear systems, the analysis of the financial markets, modelling of signals etc. Quality of work of neural networks depends from efficiency of the chosen algorithm of definition of weights of a network for achievement of required accuracy on training and test samples. The method of adjustment of weights of neural net on the basis of the theory of optimal control and representation of neural network by dynamical system is offered below.

---

### Representation of a Network by the System of Recurrent Equations

---

As is known [4], neural network - is set of the same elements - neurons, - divided in to the parts-layers consistently connected among themselves. Each of neurons from, actually, answers scalar function of vector argument  $y=F(w^T x)$ , that is superposition of the linear form with a vector of the linear form  $w$  which name as vector of weights, - and scalar function  $F$ . Function  $F$  is activation function of neuron. Vector argument  $x$  is an input of the neuron, scalar value  $y$  - an output. Inputs of the neuron belong to this or that layer on which the network is broken. These layers are ordered consistently so, that outputs of all neurons of the previous layer move on inputs of any of neurons of the following layer. An input of the first layer is the signal which is input

for whole network. For standardization of designations, we shall consider that the input forms a layer with number 0. This layer does not contain neurons and, actually, sets an input signal  $x(0)=x_0$  which consists from  $l_0$  components. The layer with number  $N$  is output. Each of layers with appropriate numbers  $k: k=0, \dots, N$  has  $l_k$  neurons. Scalar outputs of the neurons of one layer are united in one vector  $x(k), k=0, \dots, N$ , which we shall consider as output of the appropriate layer. Dimension of such vector coincides with quantity  $l_k, k=0, \dots, N$  of the neurons in the appropriate layer.

Let's consider that all neurons of the same layer have identical weights. General weight for all neurons of one layer we shall designate accordingly by number of a layer  $w(k): w(k) = (w(k)_1, \dots, w(k)_{l_{k-1}})^T, k=1, \dots, N$ . Dimension of a vector of weights, naturally, coincides with quantity  $l_{k-1}$  neurons of the layer-predecessor.

For the activation functions of any of neurons of the appropriate layer, we shall consider, that they are different for every neuron and will be designated  $F_i^{(k)}(z), i=1, \dots, l_k, k=1, \dots, N$ . We shall remind, that activation functions are scalar functions of scalar arguments.

Thus, transformation of an input signal  $x(0)=x_0$  by consecutive layers of the neural network is described by system of recurrent equations:

$$x(k+1) = \begin{pmatrix} F_1^{(k+1)}(w(k+1)^T x(k)) \\ \dots \\ F_{l_{k+1}}^{(k+1)}(w(k+1)^T x(k)) \end{pmatrix}, k=0, \dots, N-1 \dots \tag{1}$$

Having designated vector function  $(F_1^{(k+1)}(z), \dots, F_{l_k}^{(k+1)}(z))$  through  $g(z, k+1)$ , we shall rewrite (1) in more compact kind:

$$x(k+1) = g(w(k+1)^T x(k), k+1), k=0, \dots, N-1. \tag{2}$$

Taking into account that each layer of a network carries out mapping from one linear space in another according to (1) or (2), the generalized scheme of the neural network may be represented by figure 1.

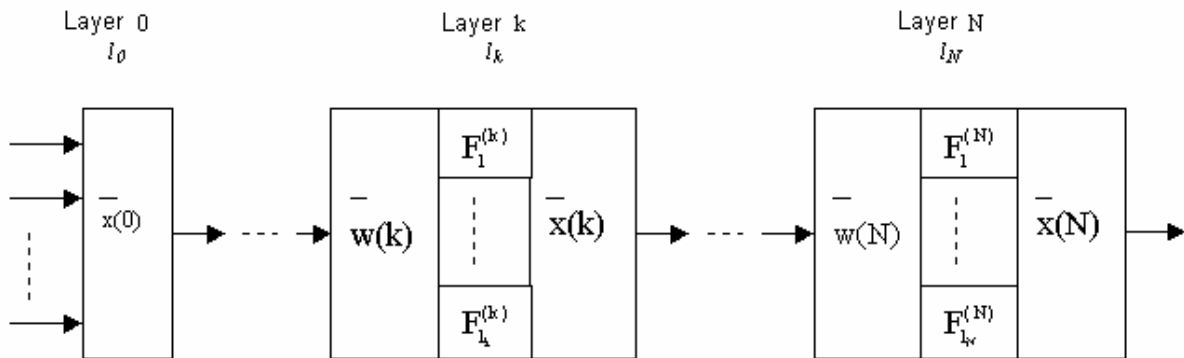


Figure 1.

Layers of the network are represented by rectangulars, which carry out transformation according to (1) or (2) in figure 1. Selected parts, in every such rectangular, which have natural interpretation:  $w(k)$  answers input synapses, parts with activation functions  $F_1^k, \dots, F_{l_k}^k$  answer neurons, the part with  $x(k)$  is responsible for concentration of outputs of neurons of a layer in a single output of the whole layer.

### The Task of Training of Neural Networks

The problem of training of the neural networks is to adjust weights  $w(k), k=1, \dots, N$  of the layers of the network so that on the given training sample: for sequences of pairs  $(x_0^{(1)}, y^{(1)}), \dots, (x_0^{(M)}, y^{(M)})$ ,  $x_0^{(i)} \in R^{l_0}, y^{(i)} \in R^{l_N}$ ,

$i = \overline{1, N}$ , in which first component is interpreted as one of variants of an input of a network and has dimension of an input layer  $l_0$ , and the second – as the desirable output of a network and has dimension of an output layer  $l$ , – the least deviation of output signals of a network from desirable was reached. Thus training of the neural networks lies in minimization of functional  $J(w(1), \dots, w(N))$ , which is determined by following equation:

$$J(w(1), \dots, w(N)) = \sum_{i=1}^M \|y^{(i)} - x^{(i)}(N)\|^2, \quad (3)$$

where  $x^{(i)}(N)$ ,  $i = \overline{1, N}$  – an output of a network for element  $i$  of a training sample:

the united output of neurons of the last layer of neural network when appropriate input value of an element of training sample moves on an input.

Let's note, that if training sample consists of one element i.e. that if a signal moves on an input, and the network should be trained on output signal, cost functional of training will be following:

$$J(w(1), \dots, w(N)) = \|y - x(N)\|^2. \quad (4)$$

---

### The Standard Approach for Solving of a Task of Training of Neural Network

---

The standard approach to training of neural network is approach according to which on the fixed input from possible variants of an input of training sample and previously fixed at any level weights of layers there is a consecutive change of these weights of each of layers in a direction opposite to a gradient of functional  $J(w(1), \dots, w(N))$  by weight of the appropriate layer  $w(k)$ ,  $k=1, \dots, N$ . Coefficients which define length of a step in the appropriate direction, should be taken rather small.

---

### Solution of a Task of Training of Neural Network by Application of Results for the Generalized Control System

---

Principle for the solution of a task of training is a statement of theorems 1, 2 below about representation of a task of training of a network by the generalized control system accordingly for: one trajectory and a beam of such trajectories.

**Theorem 1.** The task of training of neural network on one input signal represents by itself optimization task for the generalized control system, in which accordingly:

- phase variables  $x(k)$ ,  $k=0, \dots, N$  is outputs of layers with appropriate numbers;
- control  $u(k)$  with appropriate number  $k=0, \dots, N-1$  coincides with weight of layer  $k+1$  and it is determined by equation:  $u(k) = w(k+1)$ ,  $k=0, \dots, N-1$ ;
- functions  $f$ , which describe recurrent connection between values of a phase variable, are determined by functions  $g$  of outputs of layers by equations:

$$f(x(k), u(k), k) = g(w(k+1)^T x(k), k+1), k=0, \dots, N-1 \quad (5)$$

- functional  $I(u_0, \dots, u_{N-1})$  coincides with  $J(w(1), \dots, w(N))$  from (2).

The evidence of the theorem is given in [2].

Effect of the theorem 1 is the opportunity to use the theorem of a kind of gradients for the generalized control system for calculation of gradients of cost functional of training in the training of neural network tasks. [2]

**Theorem 2.** The task of training of neural network on training sample of any quantity  $M$  represents by itself optimization task for a beam of dynamics, in which accordingly:

- phase variables  $X(k)$ ,  $k=0, \dots, N$  are matrix and consist from columns  $x^{(i)}(k)$ ,  $i = \overline{1, M}$ , each of which is an output of a layer with appropriate number if appropriate input element  $x_0^{(i)}$ ,  $i = \overline{1, M}$  from training sample;

- control  $u(k)$  with appropriate number  $k=0, \dots, N-1$  coincides with weight of a layer  $k+1$  and is determined by equation:  $u(k) = w(k+1)$ ,  $k=0, \dots, N-1$ ;
- functions  $F=F(X(k), u(k), k)$ , which describe recurrent connection between values of a phase variable, are matrix and consist from columns  $(f(X(k), u(k), k))_i$ ,  $i=1, \overline{M}$ , which are determined by functions  $g$  of outputs of the appropriate layers of neural network according to equations:

$$f(X(k), u(k), k)_i = g(w(k+1)^T x^{(i)}(k), k+1), \quad k=0, \dots, N-1, \quad (6)$$

where  $x^{(i)}(k)$ ,  $i=1, \overline{M}$  – an output of a layer with number  $k$ :  $k=1, \overline{M}$ , as reaction on element  $i$  of the training sample;

- functional  $I(u_0, \dots, u_{N-1})$  coincides with  $J(w(1), \dots, w(N))$ .

*The evidence.* The evidence will be the same as for previous result and given in [2].

Effect of the theorem 2 is the opportunity to use the theorem of a kind of gradients for the generalized beam of dynamics, for calculation of gradients of cost functional of training in a task of training of neural network. [2] It, actually, is result of the following theorem.

**Theorem 3.** Gradients of cost functional of training of neural network are determined by equations:

$$\text{grad}_{w(k)} J(w(1), \dots, w(N)) = -\text{grad}_{w(k)} \sum_{i=1}^M H^{(i)}(x^{(i)}(k), w(k+1), p^{(i)}(k+1), k) \quad (7)$$

$$k=1, \dots, N.$$

Theorem 3 forms a basis of Error Back Propagation algorithm for training of neural networks.

## Conclusions

The adjustment of weights of neural networks method is described in the article. It is based on the theory of optimal control and representation of neural networks by beam of dynamics. Representation of neural networks by beam of dynamics allows to adjust weights of neural networks effectively and thus to solve a task of training.

## Literature

1. Кириченко Н. Ф., Крак Ю. В., Полищук А.А. Псевдообратные и проекционные матрицы в задачах синтеза функциональных преобразователей. // Кибернетика и системный анализ –2004.–№3.
2. Кириченко Н.Ф., Донченко В.С., Сербаяв Д.П. Нелинейные рекурсивные регрессионные преобразователи: динамические системы и оптимизация. // Кибернетика и системный анализ – 2005.–№1.
3. Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems, 2, 303–314.
4. Haykin, S. (1999). Neural networks: A comprehensive foundation (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.
5. Kohonen, T. (1995). Self-organizing map. Heidelberg, Germany: Springer-Verlag.
6. Бублик Б.Н., Кириченко Н.Ф. Основы теории управления. – К.: Высшая школа, 1975.–328 с.

## Authors' Information

**Volodymyr Donchenko** – professor, Kiev National Taras Shevchenko University, Department of System Analysis and Decision Making Theory, e-mail: [vson@unicyb.kiev.ua](mailto:vson@unicyb.kiev.ua)

**Denis Serbaev** – post graduate course student, Kiev National Taras Shevchenko University, Department of System Analysis and Decision Making Theory, e-mail: [sdp@unicyb.kiev.ua](mailto:sdp@unicyb.kiev.ua)

## AUTOMATA-BASED METHOD FOR SOLVING SYSTEMS OF LINEAR CONSTRAINTS IN $\{0,1\}$

Sergey Krivoi, Lyudmila Matvyeyeva, Wioletta Grzywacz

*Abstract:* We consider a finite state automata based method of solving a system of linear Diophantine equations with coefficients from the set  $\{-1,0,1\}$  and solutions in  $\{0,1\}$ .

*Keywords:* system of the linear Diophantine equations, set of the basis solutions, outputless finite state automaton.

*ACM Classification Keywords:* F.1.1 Models of Computation - finite automata, systems of linear Diophantine constraints

---

### Introduction

Algorithms for computing the basis of solutions of a system of homogenous linear Diophantine equations (SHLDE) in the set of natural numbers can be applied to numerous problems. These applications include the Petri nets (determining invariants, traps and deadlocks), proving contraction of a set of disjuncts, Presburgers arithmetic, logical deductions in the predicate logic, etc. The special case of SHLDE is one with solutions over the set  $\{0,1\}$  and coefficients coming from the set  $\{-1, 0, 1\}$ . In this work we present an algorithm for computing the basis of all solutions of a SHLDE of this type. We'll apply this algorithm to a general form of a SHLDE and to a system of non-homogenous inequalities and equations as well.

---

### 1. Necessary Knowledge

The system in the following format is called SHLDE:

$$S = \begin{cases} a_{11}x_1 + \dots + a_{1q}x_q = 0 \\ a_{21}x_1 + \dots + a_{2q}x_q = 0 \\ \dots \dots \dots \dots \dots \dots \dots \\ a_{p1}x_1 + \dots + a_{pq}x_q = 0 \end{cases} \quad (1)$$

where  $a_{ij} \in \{-1,0,1\}$ ,  $x_j \in \{0,1\}$ ,  $i \in [1, p]$ ,  $j \in [1, q]$ . Vectors  $e_1 = (1,0,\dots,0,0)$ ,  $e_2 = (0,1,\dots,0,0)$ ,  $\dots, e_q = (0,0,\dots,0,1)$  are the vectors of canonical basis. The vector  $c = (c_1,\dots,c_q)$ ,  $c_i \in \{0,1\}$  is the solution of SHLDE  $S$ , when following identity holds:

$$\forall i \in [1, p] \quad a_{i1}c_1 + a_{i2}c_2 + \dots + a_{iq}c_q \equiv 0,$$

where  $\text{sign} \equiv$  denotes relation of identity. Comparison of two vectors  $x = (x_1,\dots,x_q)$  and  $y = (y_1,\dots,y_q)$  from set  $N^q$  can be defined as follows:

$$x \leq y \Leftrightarrow \forall i \in [1, q] \quad x_i \leq y_i.$$

This relation is a partial order relation and the minimal vectors of the solution of the SHLDE, w.r.t. this order, are basis solution of a SHLDE. We present an algorithm of building this basis based on the theory of the finite automata.

---

## 2. Algorithm Description

---

Presented algorithm is based on the outputless finite automaton theory. To describe this algorithm first we consider the case of a single Diophantine equation and next the case of the system of equations.

---

### 2.1. Case of a Single Equation

---

Consider the equation:

$$a_1x_1 + a_2x_2 + \dots + a_qx_q = 0 \quad (2)$$

where  $a_j \in \{-1,0,1\}$ ,  $x_j \in \{0,1\}$ ,  $j \in [1, q]$ . Using the TSS algorithm for computing the set of the solutions of the equation (2) the set of basis solutions of this equation is obtained. This follows from the theorem:

**Theorem 1** If an equation  $a_1x_1 + a_2x_2 + \dots + a_qx_q = 0$  is given, where  $a_j \in \{-1,0,1\}$ ,  $j \in [1, q]$  then the TSS of this equation consists of the set of basis solutions of this equation.

*Proof:* Let  $c = (c_1, \dots, c_q)$ ,  $c_i \in \{0,1\}$  be any solution of equation (2), and its non-negative coordinates  $c_{j_1}, \dots, c_{j_q}$  be given. Lets exclude from this set the set of non – zero coordinates, all of which correspond to zero – valued coefficients of the equation (2). It means that we obtain the vector  $c' = c - e_{j_1} - \dots - e_{j_r}$ , where  $e_{j_i}$  ( $i \in [1, q]$ ) are vectors of canonical basis, which are also the elements of the TSS. Obviously the vector  $c'$  is the solution of the equation (2) as well. If  $c' = 0$  then the theorem is proved, and  $c = e_{j_1} + \dots + e_{j_r}$ . If  $c' \neq 0$ , then the number of non-zero coordinates is even and moreover one half of these coordinates corresponds to the coefficients equal to  $-1$  in the equation (2), and the second half — equal to  $+1$ . Since the TSS contains all vectors corresponding to the combinations of the coefficients equal to  $-1$  and  $1$  then indeed  $c'$  can be presented in the form of non-negative linear combination presented below:

$$c' = e_{i_1} + \dots + e_{i_t},$$

where  $e_{i_j} \in TSS$  of the equation (2). Since  $c' = c - e_{j_1} - \dots - e_{j_r}$  then  $c = e_{j_1} + \dots + e_{j_r} + e_{i_1} + \dots + e_{i_t}$ .

Theorem is proved.

**Corollary:**

- Number of the basis solutions of the equation (2) equals  $k * m + r$ , where  $k, m$  - are positive and negative coefficients of this equation respectively.
- Computing the set of the basis solution of the equation (2) is done in  $O(q^2)$  time.

**Example 1.** Let the following equation be given:

$$1x_1 + 0x_2 - 1x_3 - 1x_4 + 1x_5 + 1x_6 - 1x_7 = 0.$$

The TSS of above equation consists of the set of the basis solution in the form of following vectors:

$$\begin{aligned} s_1 &= (0,1,0,0,0,0,0), s_2 = (1,0,1,0,0,0,0), s_3 = (1,0,0,1,0,0,0), s_4 = (1,0,0,0,0,0,1), \\ s_5 &= (0,0,1,0,1,0,0), s_6 = (0,0,1,0,0,1,0), s_7 = (0,0,0,1,1,0,0), s_8 = (0,0,0,1,0,1,0), \\ s_9 &= (0,0,0,0,1,0,1), s_{10} = (0,0,0,0,0,1,1). \end{aligned}$$

Number of zero coefficients of this equation equals 10, because  $r = 1, k = 3, m = 3$ , and we get  $k * m + r = 3 * 3 + 1 = 10$  being resulted from the corollary.

## 2.2. Automata-based Presentation of a Basis Solution

Let  $A = (A, X, f, \{a_0\}, F)$  be the outputless finite automaton, where  $A$  - a finite set of states,  $X$  - a finite set of input symbols,  $f : A \times X \rightarrow A$  - transition function,  $a_0 \in A$  - starting state,  $F \subseteq A$  - a set of the terminal states. Presentation of the set of basis solutions of the equation (2) by means of the automata has the following form. Set of states of the automaton  $A = \{0, 1, -1\}$ ,  $X = \{e_1, e_2, \dots, e_q\}$ , where  $e_i$  ( $i \in [1, q]$ ), are the vectors of the canonical basis from  $N^q$ , the starting state and terminal state are the state 0, and transition function  $f$  is defined as follows:

$$f(0, e_i) = 0 \text{ if } e_i \text{ corresponds to coefficient } 0;$$

$$f(0, e_i) = 1 \text{ if } e_i \text{ corresponds to coefficient } 1;$$

$$f(0, e_i) = -1 \text{ if } e_i \text{ corresponds to coefficient } -1;$$

$$f(1, e_i) = 0 \text{ if } e_i \text{ corresponds to coefficient } -1;$$

$$f(-1, e_i) = 0 \text{ if } e_i \text{ corresponds to coefficient } 1;$$

in other cases function  $f$  is unspecified.

Example 2. For the equation from the example 1 we have:

$$A = \{0, 1, -1\}, X = \{e_1, e_2, \dots, e_7\}, a_0 = 0, F = \{a_0\},$$

and transition function  $f$  is defined in the form of transition table:

F	0	1	-1
$e_1$	1	-	0
$e_2$	0	-	-
$e_3$	-1	0	-
$e_4$	-1	0	-
$e_5$	1	-	0
$e_6$	1	-	0
$e_7$	-1	0	-

The vector of the solutions is built per the word  $p$ , which moves the automaton from the state 0 to the state 0 in the following way. If  $p = e_{i_1}e_{i_2} \dots e_{i_k}$  then vector of solutions  $x_p = e_{i_1} + e_{i_2} + \dots + e_{i_k}$ , where  $e_{i_j} \in X$ ,  $j \in [1, k]$ . It is clear that we are interested in words corresponding to the cycles in automaton, from starting state 0 to terminal state 0. For example, in the above automaton the words:

$$p_1 = e_1e_3, p_2 = e_1e_4,$$

$$p_3 = e_1e_7, p_4 = e_5e_3, p_5 = e_5e_4, p_6 = e_5e_7, p_7 = e_6e_3, p_8 = e_6e_4, p_9 = e_6e_7, p_{10} = e_2$$

creates the set of the basis solutions composed of the vectors  $s_1 - s_{10}$  (see example 1).

Theorem 2. Automaton defined in the above form presents the set of all solutions of the equation (2).

Proof follows directly from the idea of automata building. Actually all the basis vectors of the solution corresponding to the zero-coefficients are presented in the given automaton, because corresponding words are obtained by transition from the state 0 to the state 0. Then all basis solutions corresponding to combination of  $-1$  and  $1$  are presented by these words as well. Since all these vectors form the set of basis solutions of (2), the theorem is proved.



### 2.3. Case of the System of the Equations

Let a SHLDE consisting of two equations be given. We'll present the sets of the basis solutions of every equation in the form of automaton. We'll build a product from the given automata and then we'll search all the words corresponding to the paths from the state (0,0) (this state is starting and terminal state) to the same state. Given set includes all of the basis vectors of the set of solution of the SHLDE. Let us consider the following example at first.

Example 3. Let the system of equations be given:

$$S = \begin{cases} 1x_1 + 0x_2 - 1x_3 - 1x_4 + 1x_5 + 1x_6 - 1x_7 = 0 \\ 1x_1 - 1x_2 + 0x_3 + 0x_4 - 1x_5 - 1x_6 + 1x_7 = 0 \end{cases}$$

Transition table of automaton corresponding to the first equation this SHLDE is described above (see example 1), and transition table corresponding to the second equation has the following form:

$f_1$	0	1	-1
$e_1$	1	-	0
$e_2$	-1	0	-
$e_3$	0	-	-
$e_4$	0	-	-
$e_5$	-1	0	-
$e_6$	-1	0	-
$e_7$	1	-	0

Transition table of the product of automata obtained before, after excluding unreachable and deadlock states has the following form:

$f_1$	(0,0)	(0,-1)	(-1,0)	(1,-1)	(-1,1)	(1,0)	(0,1)
$e_1$	-	(1,0)	(0,1)	-	-	-	-
$e_2$	(0,1)	-	-	-	-	(0,0)	(0,0)
$e_3$	(-1,0)	-	-	-	-	-	-
$e_4$	(-1,0)	-	-	-	-	-	-
$e_5$	(1,-1)	-	(0,-1)	-	(0,0)	(1,0)	(1,0)
$e_6$	(1,-1)	-	(0,-1)	-	(0,0)	(1,0)	(1,0)
$e_7$	(-1,1)	(-1,0)	-	(0,0)	-	-	-

Starting and terminal state in this automaton is the state (0,0). The search of cycles in obtained automata results in the following words:

$$p_1 = e_2e_1e_3, p_2 = e_2e_1e_4, p_3 = e_3e_1e_5e_4, p_4 = e_3e_1e_6e_4, p_5 = e_5e_7,$$

$p_6 = e_6e_7, p_7 = e_7e_6, p_8 = e_2e_7e_1e_5e_3, p_9 = e_2e_7e_1e_6e_3$  etc. These words correspond to the following set of the basis solutions:

$$s_1 = (1,1,1,0,0,0,0), s_2 = (1,1,0,1,0,0,0), s_3 = (1,0,1,1,1,0,0), \\ s_4 = (1,0,1,1,0,1,0), s_5 = (0,0,0,0,1,0,1), s_6 = (0,0,0,0,0,1,1).$$

The analysis of the example presented above results in the following conclusion:

- the size of the automaton corresponding to the SHLDE is not bigger than  $3 * q$ ;

- the memory required for representing original automaton is proportional to  $3 * p * q$  ;
- the length of the longest cycle in product of initial automata is not bigger than  $q$  .

**Theorem 3.** The set of solutions of a SHLDE, corresponding to words representing simple cycles in an automaton being a direct product of the original automata, contains all the basis solutions of the given SHLDE.

**Proof:** Let us assume that the SHLDE consists of two equations. Assuming that  $x$  is a solution of SHLDE, then the vector  $x$  is equivalent to the word  $p$  in the automaton representing the first equation. The same vector  $x$  represents word  $p$  in the automaton representing the second equation. This is the result of the assumption that the original automata represent the set of all the possible solutions of the SHLDE. Under this assumption the word  $p$ , belonging to the common part of the two regular languages represented as a product of the original automata, should take the automaton from a starting state to a starting state (being at the same time a terminal state of this automaton). This means that word  $p$  corresponds to some cycle in the automaton. We'll represent this word as a concatenation of its subwords  $p_1, p_2, \dots, p_k$ , representing cycles. As words  $p_1, p_2, \dots, p_k$  generate the basis solutions we shall call them  $e_1, e_2, \dots, e_k$ . Then it becomes obvious that  $x = e_1 + e_2 + \dots + e_k$ .

### 3. Time Characteristics of the Algorithm

From the general automata theory we know that the number of states of an automaton that represents a set of all solutions of a SHLDE having  $p$  equations does not exceed  $3^p$ . This estimation shows us that the time required to build such an automaton, to find all the cycles and distinguish the basis solutions will be proportional to  $3^p$ , which is not very optimistic value. Obtained estimation does not contradict the result saying that the complexity of the algorithm for the basis of the set of SHLDE solutions creation will be exponential. In most cases the number of states of the resulting automaton is significantly smaller than  $3^p$ . Obviously the redundancy of the calculations in the presented algorithm comes from the fact that some of the calculations are made multiple times. For example the result (0,0,0,0,1,0,1) is obtained twice (the same as the other results) and the automaton, which transition table is presented below, presents the same set of solutions as the automaton in the example 3, but it has lesser number of states.

$f_3$	(0,0)	(0,-1)	(-1,0)	(-1,1)	(1,0)
$e_1$	-	(1,0)	-	-	-
$e_2$	(0,-1)	-	-	-	-
$e_3$	(-1,0)	-	-	-	-
$e_4$	(-1,0)	-	-	-	-
$e_5$	-	-	(0,-1)	(0,0)	(1,0)
$e_6$	-	-	(0,-1)	(0,0)	(1,0)
$e_7$	(-1,1)	-	-	-	-

It should be mentioned as well that because the solutions of a SHLDE belong to a set  $\{1,0\}$  all the vectors in the algorithm are Boolean which simplifies the calculations.

The biggest advantage of this algorithm is that the computation of the basis of the set of all the solutions does not require the explicit form of the automaton being the product of the original automata to be built, instead we create the basis directly using the original automata by going in parallel through the corresponding states of those automata. This allows for a strong memory usage optimization and as a result to obtain greater calculations efficiency. For example the creation of the explicit form of the automaton in the example 3 is not necessary.

The basis can be built using only the original automata which present the sets of solutions of the first and the second equation respectively.

Finally, as it has been shown above, the presented algorithm allows for further optimization. This possibility comes from the fact that during the creation of the basis of the set of solutions for the automaton representing the complete set of solutions many results are encountered multiple times. This redundancy comes from the fact that the concatenation in this case is associative. This presents another problem: *how to minimize finite automaton taking into account the associativity of the concatenation of the words in the input alphabet of the automaton.*

#### 4. Second Approach to the Creation of an Automaton which Represents the Basis of the Set of Solutions

We'll describe another method of the creation of an automaton representing the basis of the set of all solutions of SHLDE, which does not require computing the product of the automata.

Let  $X = \{s_1, s_2, \dots, s_n\}$  - the basis of the set of solutions of the first equation SHLDE  $S$  and  $L_2(x) = 0$  - the second equation in the system  $S$ . Using the set  $S$  and function  $L_2(x)$  we'll build a finite automaton  $A = (A, X, \{0\}, f, \{0\})$  in which the state 0 is starting and terminal state and the transition function is as follows:

$$f(a, s_i) = \begin{cases} L_2(s_i), & \text{if } a = 0; \\ a + L_2(s_i), & \text{if } a \neq 0 \ \& \ |a + L_2(s_i)| < |a|, \\ \text{unspecified} & \text{in the other case} \end{cases}$$

After creating the automaton in this way we compute the basis of the set of solutions of the subsystem  $L_1(x) = 0$  &  $L_2(x) = 0$  of the system  $S$ . Elements of this set play a part of the new alphabet. This alphabet and the function  $L_3(x)$  (if exists), being on the left-hand side of the equation  $L_3(x) = 0$ , is used to build the next automaton and so on, until we have created the basis of the set of all solutions of the SHLDE  $S$ .

Lets consider the SHLDE  $S$  from the example 3. The results of the function  $L_2(x)$  for the vectors from the set  $X = \{s_1, s_2, \dots, s_{10}\}$  are the values: -1,1,1,2,-1,-1,-1,-1,0,0. The automaton  $A$  representing the alphabet  $X$  and function  $L_2(x)$  is presented below:

$f$	0	-1	1	2
$s_1$	-1	-	0	1
$s_2$	1	0	-	-
$s_3$	1	0	-	-
$s_4$	2	-	-	-
$s_5$	-1	-	0	1
$s_6$	-1	-	0	1
$s_7$	-1	-	0	1
$s_8$	-1	-	0	1
$s_9$	0	-	-	-
$s_{10}$	0	-	-	-

We create the vector-solutions of SHLDE. These vectors correspond to the simple cycles from the state 0 into the same state 0. In our case these vectors are the following:

$$s_1 + s_2 = (1,1,1,0,0,0,0), s_1 + s_3 = (1,1,0,1,0,0,0), s_5 + s_2 = (1,0,2,0,1,0,0), s_6 + s_2 = (1,0,2,0,0,1,0),$$

$$\begin{aligned}
s_6 + s_3 &= (1,0,1,1,0,1,0), s_7 + s_2 = (1,0,1,1,1,0,0), s_7 + s_3 = (1,0,0,2,1,0,0), s_8 + s_2 = (1,0,1,1,0,1,0), \\
s_8 + s_3 &= (1,0,0,2,0,1,0), s_4 + s_1 + s_5 = (1,1,1,0,1,0,1), s_4 + s_1 + s_6 = (1,1,1,0,0,1,1), s_4 + s_1 + s_7 = (1,1,0,1,1,0,1), \\
s_4 + s_1 + s_8 &= (1,1,0,1,0,1,1), s_4 + s_5 + s_6 = (1,0,2,0,1,1,1), s_4 + s_5 + s_7 = (1,0,1,1,2,0,1), \\
s_4 + s_5 + s_8 &= (1,0,1,1,1,1,1), s_4 + s_6 + s_7 = (1,0,1,1,1,1,1), s_4 + s_6 + s_8 = (1,0,1,1,0,2,1), \\
s_4 + s_7 + s_8 &= (1,0,0,2,1,1,1), s_9 = (0,0,0,0,1,0,1), s_{10} = (0,0,0,0,0,1,1).
\end{aligned}$$

The basis vectors are:

$$\begin{aligned}
s_1 &= (1,1,1,0,0,0,0), s_2 = (1,1,0,1,0,0,0), s_3 = (1,0,1,1,1,0,0), \\
s_4 &= (1,0,1,1,0,1,0), s_5 = (0,0,0,0,1,0,1), s_6 = (0,0,0,0,0,1,1),
\end{aligned}$$

which is confirmed by the solution of this SHLDE obtained by the previous method.

---

## 5. Summary

---

Presented algorithms can be applied to a general form SHLDE, systems of non-homogenous equations and systems of linear inequalities as well. In these cases we introduce one (for the system of non-homogenous equations) or more (for the system of inequalities) additional variables thus reducing them to a problem of solving a SHLDE in the set  $\{0,1\}$ .

Given algorithms are especially effective for determination of the incompatibility of SHLDE and for computing the first solution of SHLDE. Obviously, when the SHLDE is incompatible, then the product of automata or an automaton created according the second algorithm, which represents the basis of the set of solutions of the given SHLDE, will be empty. From the method of creation of these automata we conclude that they will be created fast enough. If a first cycle appears in the process of building of such automata, it means that the SHLDE is compatible and, if necessary, the solution of SHLDE, corresponding to this cycle, can be compute and further computing can be canceled accordingly.

It should be mentioned that in the second approach the same problem of minimization of a finite automaton arises taking into account the associativity of the concatenation of the words in the input alphabet  $X$  of the automaton.

---

## Bibliography

---

- [Murata, 1989] T. Murata. Petri Nets: Properties, Analysis and Applications. In Proceedings of the IEEE, 1989, Vol. 77, No:4, pp. 541-580.
- [Baader, 1994] Baader F., Ziekmann J. Unifications Theory. In Handbook of Logic in Artificial Intelligence and Logic Programming. – Oxford University Press. 1994. pp. 1-85.
- [Common, 1999] Common H. Constraint solving on terms: Automata techniques (Preliminary lecture notes). Intern. Summer School on Constraints in Computational Logics: Gif-sur-Yvette, France, September 5-8. 1999. 22 p
- [Lloyd, 1987] Lloyd J. Foundations of Logic Programming (2-d edition). Springer Verlag. 1987. 276 p.
- [Krivoi, 2003] Krivoi S. Algorithms of solving systems of linear diophantine constraints in space  $[0,1]$ . Cybernetics and system analysis. 2003, number 5. pp. 58-69.

---

## Authors' Information

---

Krivoi Sergey – Glushkov Institute of Cybernetics NSA Ukraine, Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: [krivoi@i.com.ua](mailto:krivoi@i.com.ua)

Lyudmila Matveyeva – Institute of Cybernetics NAS of Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: [luda@iss.org.ua](mailto:luda@iss.org.ua)

Grzywacz Wioletta – Technical University of Czestochowa, Institute of Computer and Information Sciences, Czestochowa, Poland, e-mail: [wiola@icis.pcz.czest.pl](mailto:wiola@icis.pcz.czest.pl)

## VERBAL DIALOGUE VERSUS WRITTEN DIALOGUE

David Burns, Richard Fallon, Phil Lewis, Vladimir Lovitskii, Stuart Owen

*Abstract: Modern technology has moved on and completely changed the way that people can use the telephone or mobile to dialogue with information held on computers. Well developed "written speech analysis" does not work with "verbal speech". The main purpose of our article is, firstly, to highlights the problems and, secondly, to shows the possible ways to solve these problems.*

*Keywords: data mining, speech recognition, natural language processing*

*ACM Classification Keywords: I.2.7 Natural Language Processing: Text analysis, speech recognition.*

---

### Introduction

---

There are several problems, which distinguish Verbal Dialogue (VD) from Written Dialogue (WD).

**Problem 1.** *Sensible VD restricts us to include in our reply to a user enquiry a very restricted number of alternative (3, 4, or maximum 5) choices, because we should take into account our restricted ability to store information temporarily in our memory [1]. For example, assume the user is searching for "Security Services Company in UK". Any information retrieval engine (e.g. Google) will assumes a maximal matching between enquiry words and documents. In the search result user almost immediately has got 14,600,000 documents and now it is the users' problem to identify the most appropriate one. In the case of VD such an approach is unacceptable even for the first 10 documents.*

**Problem 2.** *Wrong recognition of Verbal Enquiry (VE) (only speaker independent speech recognition is considered). As distinct from WD VD does not have any misspelling problem (we assume that grammar includes only correctly spelled words) because any VE will be converted into a sequence of "correct words". For example, a user said his postcode is: "PL3 4PX ". There are a minimum of 4 different possible results of recognition of this VE: (1) "PL3 4PX "; (2) "BL3 4PX"; (3) "PL3 4BX"; (4) "BL3 4BX". The last two results: (3) and (4) represent non existing postcodes and Natural Response (NR) system [2] will simply ask user to repeat postcode. In the case of the second result "BL3 4PX" represents existing postcode but wrong result.*

**Problem 3.** *Barging (VD interruption) is absolutely natural style of VD. This is a fragment of VD:*

NR:           *Where would you like to go?*  
 User:         *The best beaches, of course.*  
 NR:           *OK, We can offer you Bahamas beach or Mia...*  
 User:         *Bahamas, please.*

The solution of this problem is more technical than scientific regarding the natural language processing and therefore will not be considered in this paper.

**Problem 4.** *TD verification and mining (TDVM). This problem is not just VD problem but is also a problem of WD. Real customer's DB's very often contain product names like: "Wht thk slcd brd" which is intended to mean "White thick sliced bread", or "Dietary spec gltn/fr & wht/frchoc/orge waf x5". A good TDVM analogy is to a body scanner. TDVM software can scan text and identify things that need to be looked at; it will show where there are anomalies in the TD. There is some specific feature of TDVM for VD: TDVM should provide selection of phonetically similar words to try to avoid the use of similar sounding words with different meaning.*

**Problem 5.** *Grammars creation. Speaker independent speech recognition has to be 'primed' around a particular vocabulary of terms and phrases, known as Grammars. To achieve reasonable recognition accuracy and response time multiple grammars should be generated automatically as a result of text data (TD) analysis. This overcomes a conflict that exists within speech recognisers that limits the effective vocabulary size without impairing the quality of recognition (normally, quality reduces as vocabulary size increases). Here we mean text*

data from an Application Domain (AD), which is represented by a database (DB). Let us distinguish two classes of grammars: independent and AD dependent grammars. Examples of independent grammars are *YesNo*, *Date*, *Time*, *Currency*, *CreditCard*, etc. Here we will discuss problem related only to AD dependent grammars creation. A grammar's "accuracy" depends on the solution of the previous problem.

## VD: Alternative Choice Minimizing

The Alternative Choice Minimizing Mechanism (ACMM) is involved only when VD problem 1 arises i.e. when in reply to VE the number of "alternative choices" exceeds some pre-specified amount of permissible alternatives (APA). APA depends on the contents of AD values e.g. if values are represented by single words then APA might equal 5: "What colour do you prefer blue, brown, white, magnolia or crème?" If values are represented by long sequences of words APA would be better assigned as 3: "We've got Mothers pride plain medium white bread 800g, Kingsmill square cut medium sliced white loaf 400g, or Hovis classic cut medium white bread 800g. Which do you prefer?"

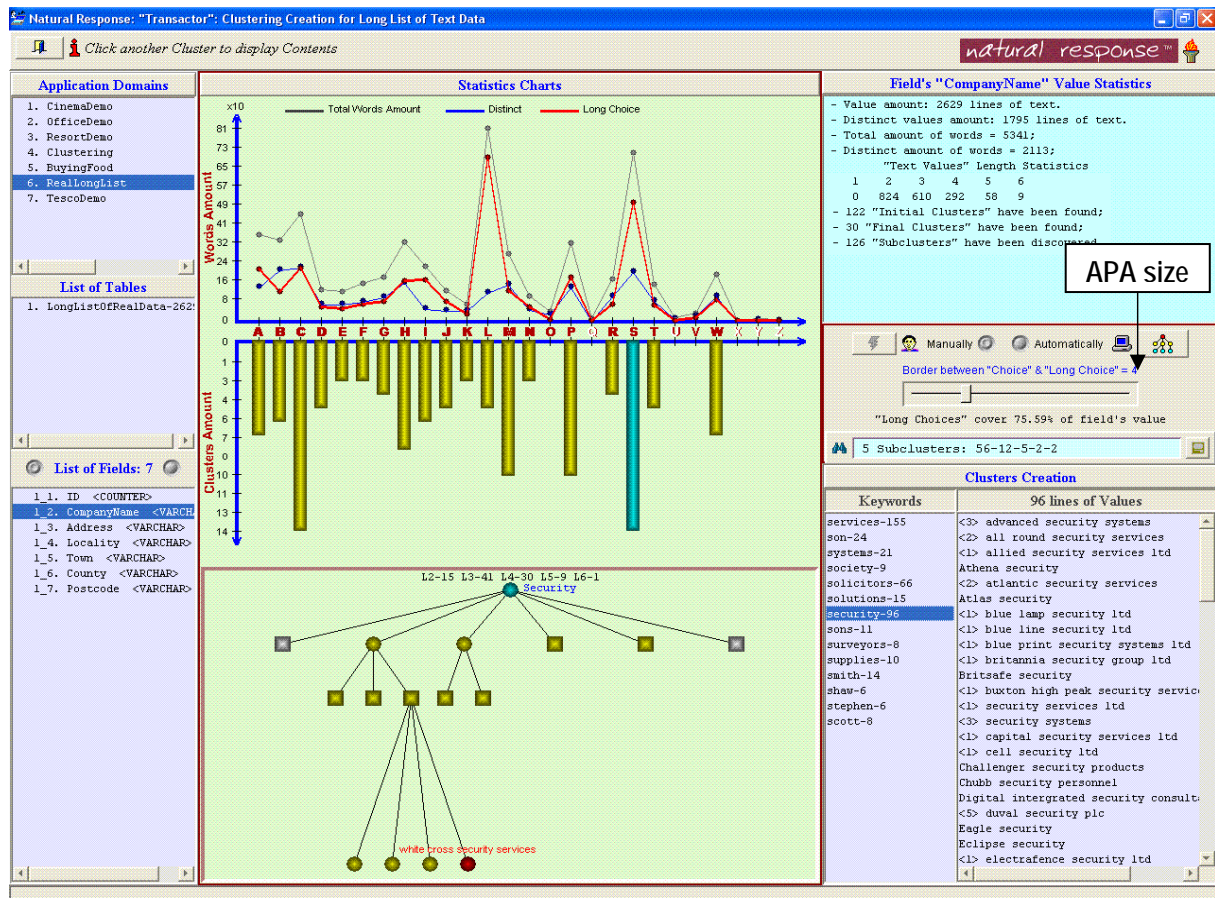


Figure 1. "Companies-Addresses" Application Domain

The best way to describe the solution to Problem 1 is to use some AD and show step by step the method of the solution. An AD database "Company-Addresses" with 2,629 companies is used (Figure 1). Knowing the TD, the user can assign an appropriate APA size before they start to analyse TD and create clusters.

To show the role of ACMM let us compare two VD.

User: I'm looking for a security company address.

NR: There are 96 different security companies. Could you please be more precise?

and User: Please help me to find a security company address.

- NR: *We found for you Security services, Security systems, Blue security or Security group. Any good for you?*
- User: *I'm afraid, not.*
- NR: *Ok, what about Security Management, Security solutions, or Reliable security. Is it better?*
- User: *Oh, yes, thank you. Security solutions, please.*
- NR: *There are Perimeter Security Solutions Ltd. and Security Solutions (Northern) Ltd. Please make your choice.*
- User: *Northern.*
- NR: *Ok, fine. The address of Security Solutions (Northern) Ltd. is Unit 7/8, Bolton, Lancashire, BL1 8TU.*

If user could not select anything suitable (e.g. instead of select "Security Solutions" user said "No") NR will continue dialogue:

- NR: *Never mind. Let's try another way. In which city you are looking for Security Company?*
- User: *Bolton.*
- NR: *We have Blue Print Security Systems Ltd, Cell Security Ltd, Duval Security Plc or Jaymac Security Products Ltd. Any good for you?*
- User: *Yes, thank you. I'm looking for Duval Security.*
- NR: *Address of Duval Security Plc is 602 Chorley New Road, Bolton, Lancashire, BL6 6EX.*

The difference between 1<sup>st</sup> and 2<sup>nd</sup> VD is provided by ACMM. The main features of ACMM are:

- ◆ As a result of TD analysis a Pyramidal Structure of Clusters (PRC) is created [3] (fragment of PRC is shown in Figure1). The criterion for each "parent node" of PRC is very simple: "striving for having no more than APA child nodes" (APA = 4). Let us distinguish three possible amounts of cluster/sub-clusters nodes (NA): (1)  $NA \leq APA$  e.g. for sub-cluster "Blue Security" there are just three members {"Blue Lamp Security Ltd", "Blue Line Security Ltd", "Blue Print Security Systems Ltd"}; (2)  $NA \leq APA * APA$ . Assume there are APA sub-clusters and each of them has no more than APA nodes. In that case replay includes "sub-cluster name" instead of "company name" e.g. "Security services", "Security systems", "Blue security or "Security group"; (3)  $NA > APA * APA$ .
- ◆ In case (1), if the user is looking for "Blue Security" NR immediately offers him/her three companies (see above). Case (2) is represented in given VD.
- ◆ As for case (3) supplementary field (or fields) (SF) are involved. In the considered VD "City" is selected as SF. SF is defined at the stage of Scenario of Dialogue creation. In real AD "Companies-Addresses" except "City" SF "Postcode" is used.

### Wrong Recognition of Verbal Enquiry

Let us consider two different variants of wrong VE recognition. Assume that in the result of TD analysis vocabulary  $\mathcal{V}$  of meaningful words and set  $\mathcal{S}$  of words sequences (each of which represents a TD line) have been created, and then grammar  $\mathcal{G} = \mathcal{V} \cup \mathcal{S}$  has been generated. First of all let us consider the situation when  $\mathcal{G} = \mathcal{V}$ . In this case when the user said word  $w_u$  speech recogniser (SR) using  $\mathcal{G}$  produced  $w_r$ . Theoretically there are four different results of such event:

- (1)  $w_u \in \mathcal{V}$  and  $w_u = w_r$  (naturally,  $w_r \in \mathcal{V}$ . But equation  $w_u = w_r$  requires some explanation. In the set of recognition results  $w_i$  and  $s_j$  are distributed in between three intervals in accordance with their scores: "Yes", "Clarifying Question" and "Don't understand" intervals.  $w_r$  denotes the top scoring word from "Yes" interval);
- (2)  $w_u \in \mathcal{V}$  and  $w_u \neq w_r$ ;

- (3)  $w_u \notin \mathcal{V}$  and  $w_u \neq w_r$   
 (4)  $w_u \notin \mathcal{V}$  and  $w_r = nil$  ( $w_r = nil$  means that  $w_r$  belongs to the "Don't understand" interval).

When  $\mathcal{G} = \mathcal{V} \cup \mathcal{S}$  and VE is represented by a sequence of words e.g.  $s_u = w_{u1} \rightarrow w_{u2}$  the situation of VE recognition is more complicated. We are not going to discuss here results of recognition for all possible combinations of input VE. We just consider a couple of the most significant cases:

- (5)  $w_{u1} \in \mathcal{V}$ ,  $w_{u2} \in \mathcal{V}$ ,  $s_u \in \mathcal{S}$ ,  $w_{u1} \neq w_{r1}$ ,  $w_{u2} \neq w_{r2}$  and  $s_u \neq s_r$ ;  
 (6)  $w_{u1} \in \mathcal{V}$ ,  $w_{u2} \in \mathcal{V}$ ,  $s_u \in \mathcal{S}$ ,  $w_{u1} \neq w_{r1}$ ,  $w_{u2} \neq w_{r2}$  and  $s_r = nil$ ;  
 (7)  $w_{u1} \notin \mathcal{V}$ ,  $w_{u2} \notin \mathcal{V}$ ,  $s_u \notin \mathcal{S}$ ,  $w_{u1} \neq w_{r1}$ ,  $w_{u2} \neq w_{r2}$  and  $s_u \neq s_r$  (but  $s_r \in \mathcal{S}$ ).

Cases (1) and (4) are obvious. For the solutions of cases (2) and (3) it is better to show on example. Suppose we have AD "Operator", which is the alternative to a receptionist, providing customers call reception. Assume the next VD took place:

- User: Can I speak to Vlad? (where  $w_u = Vlad$ . SR recognised Vlad as Wlad from "Yes" interval, i.e.  $w_r = Wlad$  and  $w_r \in \mathcal{V}$  - case (2)).  
 NR: Certainly, connecting you now (i.e. connecting to the wrong person).

**Solution.** To avoid annoying "clarification questions" e.g. "Did you say Vlad?", recognised word from "Yes" interval is accepted as a "correct word" but such an approach might be cause a wrong result of VD. To sort out this problem a set of heuristics production rules has been developed to check words from the "Yes" interval e.g. "If 1<sup>st</sup> letter of  $w_r$  belongs to  $\{b, p, v, w, \dots\}$  ask "clarifying question".

Case (3) is more complicated. Suppose that the position of someone from the customer's company (AD "Operator") is not "Director" but instead "Chief Executive Officer" been used. AD "Operator", as well as connecting users, can navigate them i.e. describe the direction to the company's office. Another words, "director"  $\notin \mathcal{V}$  and "direction"  $\in \mathcal{V}$ . The next VD might take place:

- User: Can I speak to a director? (where  $w_u = director$ . SR recognised director as direction from "Yes" interval, i.e.  $w_r = direction$  and  $w_r \in \mathcal{V}$ ).  
 NR: With pleasure. Where are you now? (completely wrong VD).

**Solution.** When AD created it is not enough to include in the knowledge base (KB) of AD only meaningful TD from customer's DB. It is important to extend text values by adding corresponding synonyms. First of all let us distinguish "closed" and "open" TD. For example, in AD "Operator" field "Name" represents *closed* data because caller should pronounce a name exactly from the existing list of Names. But field "Position" represents *open* data because caller might say: "I would like to talk with a director (chairman, boss, managing director etc.)". In the case of considered field "Position" the solution is very simple because the set of valid positions in UK is well defined and very restricted. It is enough just to provide a link between the existing position i.e. "Chief Executive Officer" and set of synonyms i.e. {director, chairman, boss, managing director etc.}. Now the previous VD will be changed:

- User: Can I speak to director?  
 NR: Our Company does not have a director position. We have chief executive officer instead. Would you like to speak to him?

To find an appropriate set of synonyms for an open field is not easy when field represents TD like "Product" containing thousands and thousands of different products. On the one hand, it is not difficult to find synonyms for meaningful words of product using, for example, WordNet [6]. WordNet is a lexical database for the English



language. Using WordNet's dictionary makes it easy to extract synonyms and meanings for any word e.g. *spring* (see Figure 2). But on the other hand, it is very difficult to automatically select appropriate synonyms for the current AD. To solve this problem the list of VE regarding AD should be analysed.

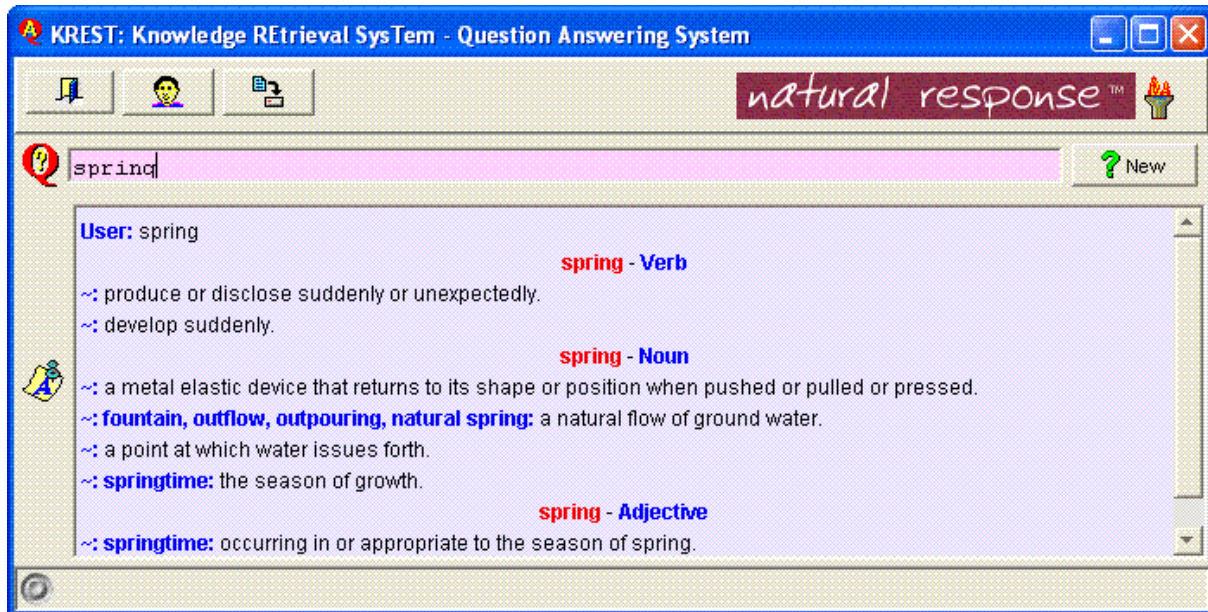


Figure 2. Synonyms and meaning of word "spring"

In the result of VE processing a language model (LM) of VE will be created. LM contains statistical information about which words or word sequences are used mostly by callers of AD. In addition to the statistical information, words adjacent within VE are analysed. The idea of adjacent words is based on the assumption that with the successive presentation of a number of words the strongest relation in it is the relation between the nearest neighbour words. "Their succeeding one after another presents evidently an important condition of structuring" [4, p.231]. It is important to underline that this information is always AD dependent. LM is stored in Sequential/Simultaneous Structure (SSS) [5]. LM of VE contains not only a set of synonyms but also patterns of VE. An example of a pattern for AD "Operator" is:

$$(\textit{speak} \oplus \textit{talk} \oplus \textit{connect}) \rightarrow \langle \textit{Name} \rangle \oplus \langle \textit{Position} \rangle;$$

where  $\oplus$  - denotes "exclusive OR";  $\rightarrow$  means "followed by";  $\langle \textit{Name} \rangle$  and  $\langle \textit{Position} \rangle$  denote any value from DB fields "Name" and "Position" respectively. Words *speak*, *talk* and *connect* (such words we call "action words") are included in  $\mathcal{V}$  along with words from fields "Name" and "Position". If a pattern of VE is available the initial VD will be have in a more "natural way" (even if word *director* is not added to  $\mathcal{V}$ ):

- User: *Can I speak to a director?* (SR recognised *director* as *direction* and correctly recognised word *speak* i.e. both *speak* and *direction* have been placed in the same "Yes" interval. Action words have priority and that is why the corresponding VE pattern has been involved to continue VD)
- NR: *Do you know the name of who you want to speak to?*
- User: *I'm afraid, not.*
- NR: *Ok, in that case would you like to talk with chief executive officer?*
- User: *Sorry, I change my mind. Can I speak to marketing manager?*
- NR: *Certainly, connecting you now.*

To discuss cases (5), (6) and (7) AD "Postcodes", which contains postcodes and cities, will be used. There are 1,741,488 postcodes in the UK. Each Postcode consists of two parts. The first part is the Outward Postcode, or Outcode. There are 2,918 outcodes. Outcode is separated by a single space from the second part, which is the

Inward Postcode, or Incode. There are 4,000 incodes and 27,089 cities and towns in UK. Postcodes recognition is very important because as soon as a postcode has been recognised correctly a very restricted set of full addresses (~30 – 60) is allocated. In order to provide postcode recognition, grammar  $\mathcal{G} = \mathcal{V} \cup \mathcal{S}$  must be generated but the size of  $\mathcal{G}$  is huge (i.e.  $|\mathcal{G}| = 1,741,481 + 2,918 + 4,000 = 1,748,399$  items) and modern SR cannot provide reliable and acceptable recognition for such size of grammar. If we exclude postcodes from  $\mathcal{G}$  and add cities it is possible significantly reduce grammar's size i.e.  $|\mathcal{G}| = 2,918 + 4,000 + 27,089 = 34,007$  items but increase "intelligence of VD". Below several VD are represented to show the solution of cases (5), (6) and (7).

User: *PL3 4PX.*  
 NR: *You said PL3 4BX. Is it correct?*  
 User: *No.*  
 NR: *Never mind. Let's try another way. Is outcode PL3 correct?*  
 User: *Yes.*  
 NR: *Ok, fine. In that case you said PL3 4PX (because only three different incodes with X link to PL3: 4PX, 5PX and 6PX). Is it right?*  
 User: *Absolutely.*

The next VD is more complicated. Town and city are involved to clarify the wrong result of postcode recognition.

User: *PL11 3BX.*  
 NR: *It is not right postcode. Did you really say BL11 3BX?*  
 User: *No.*  
 NR: *No matter. Let's try another thing. Tell me your town, please.*  
 User: *Seaton.*  
 NR: *There are 8 towns Seaton in different areas like Exeter, Sunderland, etc. What is your nearest city?*  
 User: *Plymouth.*  
 NR: *Great. The outcode of Seaton in Plymouth area is PL11. Is it correct?*  
 User: *Of course.*  
 NR: *Consequently you expressed your postcode as PL11 3BX. Isn't it?*  
 User: *Oh, at last. I really said so.*

---

### Text Data Verification and Mining

---

By default two different kind of DB exist in reality – with correct TD and "wrong" TD. For example, DB, which contains TD about *people, addresses, companies* etc. as a rule have correct TD and require minimum effort for TD verification. Result of TDVM for such TD is shown on Figure 1. The typical example DB with "wrong" TD is any DB with "Product name" in it. Let us consider the real AD "Product" (Figure 3). We will call TD "wrong" if they contain *wrong words* (WW) and/or *unknown words* (UW). Two different kind of WW will be distinguished:

(1) abbreviation, which is a sequence of consonants letter. For example, "Wht thk slcd brd" means "White thick sliced bread". (2) Sequence of symbols when sequence includes not only letters but also some symbols e.g. "Dietary spec gltn/fr & wht/frchoc/orge waf x5" (We don't know the meaning of that).

The definition of UW is very simple: Let us call word UW if it absent in Oxford dictionary. There are three possible reasons for it: (1) word shortening e.g. "org" for *organic* (org was used in 261 names of product e.g. "Doves Farm Org Lemon Cookies Non Gluten"); (2) new words (= brand) e.g. *Nestle, Cadburys* etc.; (3) misspelling. Result of DTVM of AD "Product" is shown on Figure 3.

At the stage of TDVM several tasks need to be solved:

- ◆ Quantity analysis – count different kinds of frequencies (see Figures 1 and 3);
- ◆ Clustering – to provide the classification of information into clusters.
- ◆ Data Verification of TD i.e. extracting WW and UW (see Figure 3). For AD “Product” 76.55% represent wrong data.
- ◆ Production Rules (PR) creation. It is very important to underline the fact that a customer’s DB cannot be corrected or reconstituted. *It means that conversion of correct user’s enquiry to wrong customer’s data needs to be provided*. For such conversions PR need to be created. Antecedent of PR might be a single word or sequence of words. Several different WW or UW might represent consequent of PR. (+) stands for exclusive OR (see PR on Figure 3).
- ◆ Data Reconstitution needs to be provided at the step of KB creation when the link between TD and DB Field is made. At this stage each product description is converted using PR from wrong TD to correct TD.

The screenshot displays the 'Data Verification' application interface. At the top, there's a search bar with 'natural response' and a 'Type in an Enquiry' prompt. The main workspace is divided into three panes: 'Tables' (showing 'tblProduct'), 'Fields' (listing various product attributes like fldPG\_Code, fldProductGroupDescription, etc.), and a list of 36,462 lines of data. Below this, a status bar indicates '77.49% of wrong data'. The bottom section contains three panes: '2,600 Wrong Words' (listing words like ch, lds, ff, etc.), '7,497 New Words' (listing words like promo, occ, gby, etc.), and '44 Production Rules for "Wrong" Words' (showing rules like 'tomatoes => tomato(+)+tomatoes(+)+tomte'). A query editor at the bottom shows an SQL query for 'white thick sliced bread'.

Figure 3. Text data verification and mining

User’s Enquiry Conversion. For an explanation it is better to consider some examples. Assume user’s enquiry was “White thick sliced bread”. Four appropriate PR have been allocated (“white  $\Rightarrow$  wht”, “thick  $\Rightarrow$  thk”, “sliced  $\Rightarrow$  slcd  $\oplus$  sld”, and “bread  $\Rightarrow$  brd”). Customer’s data might contain any combination of wrong and correct words.

Theoretically for the considered example there are 16 possible combinations of data, namely: (1) "White thick sliced bread", (2) "White thick sliced brd", ... , (16) "wht thk (slcd ⊕ sld) brd". Result of such conversion is shown on Figure 3.

### Grammar Creation

Let us consider three aspects of grammar creation. The first one is regarding the ability of SR to better recognise (i.e. more reliable, with higher score) a sequence of words than single words e.g.  $G = \{white\ bread\}$  provides better recognition of VE "white bread" than  $G = \{white, bread\}$ . But for reliable recognition it is not enough to include in  $G$  just product names because the user is absolutely free to ask whatever he/she wants about, for instance, "White thick sliced bread", namely, "white bread", "sliced bread", "thick sliced bread" etc. On the one hand SSS (Sequential/Simultaneous Structure [5]) allows us to generate sensible sequences, on the other hand, the size of  $G$  is critical. A complex grammar usually causes a less accurate recognition because of the larger number of possible word sequences. The solution is to extract frequently used utterances from the users' VE. The important point is – any NR system, which provides VD, should be a self-learning system. Indeed, during the working day thousands and thousands of telephone calls to AD "Product" are taking place. "Sleeping time" is more than enough to update: (1) set of synonyms; (2) enquiry patterns and (3) grammar.

*(white\_sliced\_bread) {white, sliced, bread, wht, slcd ⊕ sld, brd}.*

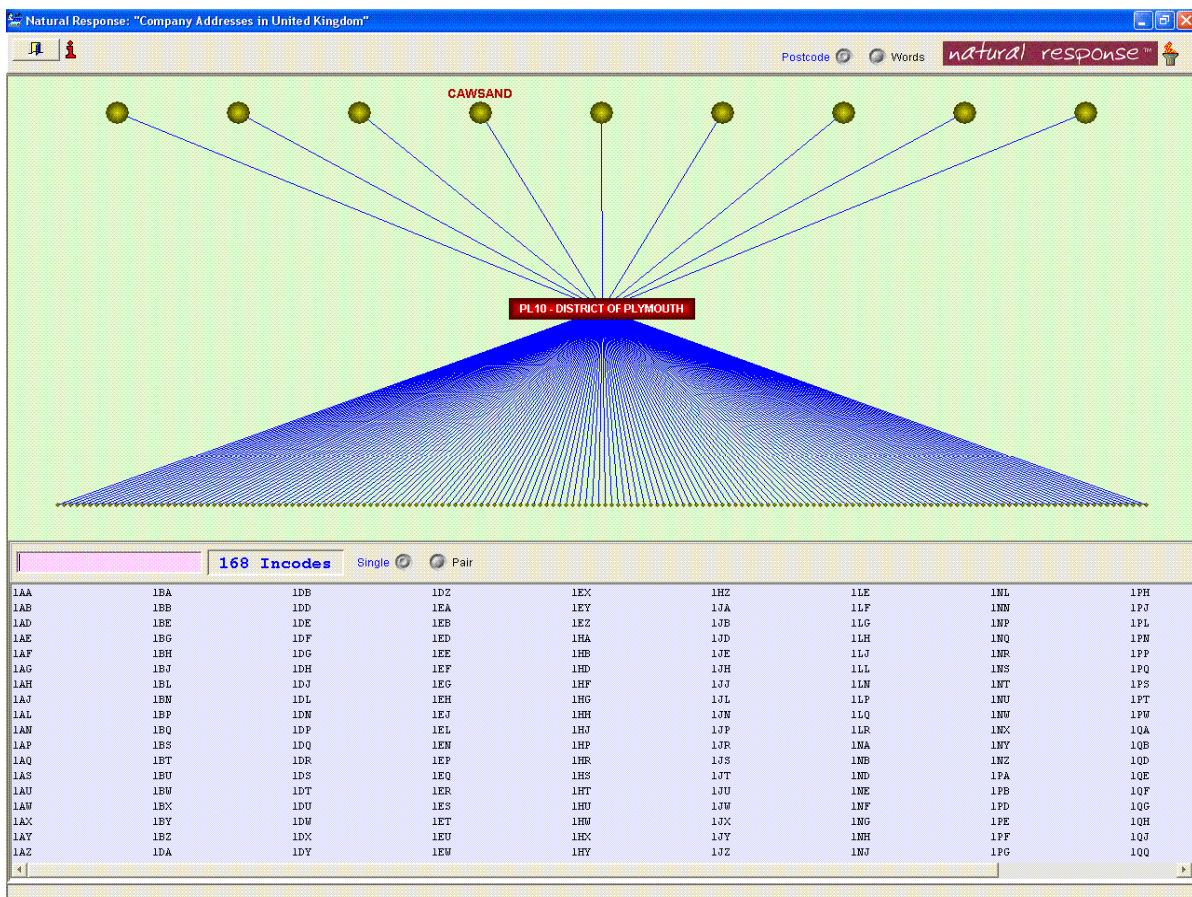


Figure 4. Towns and Incodes allocation



The second aspect is regarding the NR reaction. It goes without saying that VD must be in real time. Any delay of VD makes NR unacceptable. The obvious way of VE processing is: (1) recognise spoken utterance; (2) using PR add WW and UW; (3) convert VE+WW+UW to corresponding SQL query; (4) run SQL query; (5) synthesize reply; (6) provide text-to-speech conversion. We can avoid step (2) and include PR in grammar. Any line of grammar might contain two parts: (*spoken word or utterance*) {*result, or tag*}. The tag result is interpreted as a string and is delimited by braces "{}". All characters within the braces are considered as parts of the tag, including white space. The line of  $G$  for utterance *white sliced bread* will look as

The third aspect is devoted to dynamically created grammars. Assume user said his outcode *PL10*. Now instead of using the full size grammar (= 4,000 items) for Incodes it is better to allocate a set of Incodes regarding *PL10* (see Figure 4) and create a grammar just with 168 items.

---

## Conclusion

---

VD and WD are "*the two sides of a coin*", the features of which should complement each other. The main purpose of our report was to call attention of scientists to distinctive features of VD. As we know from our scientific experience it is always easier to find the best solution when you have an access to some practical results. That is why we included in our report real results but not ideal, or "dream" ones.

---

## Bibliography

---

- [1] G.A. Miller. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity to Process Information. *Psy.Rev.*,63,81-97,1956.
- [2] Natural Response™. [www.naturalresponse.net](http://www.naturalresponse.net), 2002.
- [3] T.M. Khalil, V. Lovitsky, "A Structure of Memory in Concept Formation", Proc. of the IEEE Conference on *Systems, Man and Cybernetics*, Boston, Mass,509-512,1973.
- [4] J.Hoffmann, Das Aktive Gedachtnis. Psychologische Experimente und Theorien zur Menschlichen Gedachtnistatigkeit, *VEB Deutscher Verlag der Wissenschaften*, Berlin, 1982.
- [5] В.А.Ловицкий, Классификация структур памяти, *Проблемы бионики*, Харьков,8,138-153,1972.
- [6] WordNet®. <http://wordnet.princeton.edu>.

---

## Authors' Information

---

David Burns – Natural Response (Natural Response is a trading style and trademark of RSVP Dialogue Limited), The i-zone, Bolton Institute, Dean Road, Bolton, Great Manchester, BL3 5AB, United Kingdom, e-mail: [david@naturalresponse.net](mailto:david@naturalresponse.net)

Richard Fallon – Natural Response, The i-zone, Bolton Institute, Dean Road, Bolton, Great Manchester, BL3 5AB, United Kingdom, e-mail: [richard@naturalresponse.net](mailto:richard@naturalresponse.net)

Phil Lewis – Natural Response, The i-zone, Bolton Institute, Dean Road, Bolton, Great Manchester, BL3 5AB, United Kingdom, e-mail: [phil@naturalresponse.net](mailto:phil@naturalresponse.net)

Vladimir Lovitskii – Natural Response, The i-zone, Bolton Institute, Dean Road, Bolton, Great Manchester, BL3 5AB, United Kingdom, e-mail: [vladimir@2ergo.com](mailto:vladimir@2ergo.com) , [vladimir@naturalresponse.net](mailto:vladimir@naturalresponse.net)

Stuart Owen – Natural Response, The i-zone, Bolton Institute, Dean Road, Bolton, Great Manchester, BL3 5AB, United Kingdom, e-mail: [stuart@naturalresponse.net](mailto:stuart@naturalresponse.net)

## LEARNING TECHNOLOGY IN SCHEDULING BASED ON THE MIXED GRAPHS

Yuri Sotskov, Nadezhda Sotskova, Leonid Rudoi

*Abstract:* We propose the adaptive algorithm for solving a set of similar scheduling problems using learning technology. It is devised to combine the merits of an exact algorithm based on the mixed graph model and heuristics oriented on the real-world scheduling problems. The former may ensure high quality of the solution by means of an implicit exhausting enumeration of the feasible schedules. The latter may be developed for certain type of problems using their peculiarities. The main idea of the learning technology is to produce effective (in performance measure) and efficient (in computational time) heuristics by adapting local decisions for the scheduling problems under consideration. Adaptation is realized at the stage of learning while solving a set of sample scheduling problems using a branch-and-bound algorithm and structuring knowledge using pattern recognition apparatus.

*Keywords:* Scheduling, mixed graph, learning, pattern recognition.

*ACM Classification Keywords:* F.2.2 Nonnumerical algorithms and problems: sequencing and scheduling.

---

### Introduction

---

Scheduling is defined as an optimal assignment of the limited resources (machines) to competitive jobs over time. Having arisen from practical needs, scheduling theory attracts much attention of management workers and operations research practitioners. One of the most general scheduling problem may be described in a framework of the multistage system with machine set  $M$  including both different machines and identical machines that should process the given set of jobs  $J = \{J_1, J_2, \dots, J_n\}$ . Each job  $J_r$  consists of the ordered set of operations. Processing time  $p_i$  and type of machine from set  $M$  that has to process operation  $i$  are assumed to be known before scheduling. Let  $M = \bigcup_{k=1}^w M_k$  where  $M_k$  denotes a set of identical machines of type  $k$ . If  $\mu_i \in M_k$  and  $\mu_j \in M_l$  with  $k \neq l$ , then machines  $\mu_i$  and  $\mu_j$  are of different types. If  $k = l$ , machines  $\mu_i$  and  $\mu_j$  are identical. We assume that operation preemptions are not allowed. As usual, each machine cannot process two (or more) operations simultaneously. The objective is to assign all the operations to suitable machines from set  $M$ , and to define for each machine  $\mu_j \in M$  the sequence of operations assigned to  $\mu_j$  in such a way that the value of the given objective function is minimal.

The complexity research has shown that even very special cases of the above scheduling problem (e.g., problems with either two different or two identical machines, or problem with three jobs and three different machines) are NP-hard [16]. Therefore, the use of exact scheduling algorithms is very limited in practice. The general methodology for developing exact scheduling algorithms is based on the branch-and-bound (B&B) method [4, 14, 16]. The exhausting enumeration while *branching* set of feasible schedules ensures the optimality of the best schedule constructed. The lower and upper *bounds* of the given objective function allow eliminate schedules, which are dominated, and reduce considerable enumeration process. However, we are forced to recognize that current improvements of the B&B algorithms cannot change radically the limit on the size of the scheduling problems solvable within reasonable running time. Moreover, it is unlikely to construct fast exact algorithms and even approximate ones with good performance measure for scheduling problems with real-world size. To be more application-oriented, different heuristics have been developed to find near-optimal feasible schedules.

It should be noted that the reasons for applying certain heuristics in dispatching algorithms are usually weakly grounded and are based mainly on the experiments. The choice of heuristics in the knowledge-based system is more adequate and it is usually computer-aided. In article [1], it is shown that a better solution can be often obtained due to appropriate changing the set of dispatching rules. As a result, the obtained processing system became better than that obtained using only a fixed set of dispatching rules. In article [12], it is demonstrated that

machine learning produces an improvement in the performance of the processing system when compared to the traditional method of using a fixed set of dispatching rules.

Local decision rules are the "backbone" for heuristic algorithms. The outcome of scheduling algorithm depends on the underlying strategy. As it was mentioned in [19], artificial intelligence provides often a better basis for modeling and solving a scheduling problem. Many real-world scheduling problems seem to be solved by semi-logical methods, such as recognizing one of a thousand familiar patterns applying to current situation and recalling the appropriate thing to do when that pattern occurs. In article [6], it was shown that by combining a learning system with simulation, a manufacturing control system can be developed that learns from its historical performance and makes its own scheduling and control decisions by simulating alternating combinations of different dispatching rules. The objective of the research described in [8] was to study the feasibility of automatically scheduling multi-machine complexes and adjusting the schedule on a real-time basis by a unified computer system. In article [2], a probabilistic learning strategy has been developed based on the principles of evolutions. In [10,17,18], it was shown how an algorithm could be extended to consider local decision rules of any number and complexity. The resulting programs do better than that based on the deterministic approach. In article [3], it is emphasized that meta-heuristics are general combinatorial optimization techniques, which are designed with the aim of being flexible enough to handle as many combinatorial problems as possible. In the recent years, these techniques have rapidly demonstrated their usefulness and efficiency in solving different combinatorial problems including scheduling problems.

Reviews of the learning-based scheduling approaches are presented in [8,11,19]. A common approach to scheduling set of the given jobs in practice is based on different dispatching rules. The performance of these rules depends on the state of the system at each moment, since there is no single rule that is better than others in all the possible system states. So, it is useful to look for the most appropriate dispatching rule at each state of the system. To achieve this goal, a scheduling approach based on machine learning seems to be rather promising. Due to analysis of the previous performance of the system, it is often possible to get knowledge that can be used to decide which dispatching rule is the most appropriate at this or that state of the system under consideration.

In this paper, exact algorithm, heuristics and pattern recognition apparatus are used in common. Our approach may be considered as a generalization of the algorithm proposed in [13] for the very special case of the above scheduling problem when all machines in set  $M$  are different (job shop). We use mixed graph model [15] along with so-called conflict resolution strategy [14–16]. The main issue of this paper is a classification scheme for local conflicts due to computational results of the exact and approximate versions of the B&B algorithm. In the next section, we formulate the scheduling problem in terms of the mixed graph model and give an overview of the conflict resolution strategy being a basis for different exact, approximate and heuristic scheduling algorithms. Modules 1, 2 and 3 of the knowledge-based system are briefly discussed in the next three sections. Some remarks are given in Conclusion.

---

### Mixed Graph Model

---

A multistage system with both different and identical machines may be described via mixed graph  $G = (V, A, E)$  [15], which is an appropriate model for constructing various scheduling algorithms [14,16]. In such a model, operation set  $V$  is represented as a set of vertices  $V = \{1, 2, \dots, n\}$ , a weight prescribed to each vertex  $i \in V$  being equal to the processing time  $p_i$  of operation  $i$ . As usual, precedence constraints between operations in set  $V$  are

represented via arc set  $A$ . Let equality  $V = \bigcup_{k=1}^w V_k$  hold, where  $V_k$  denotes the set of all operations processed by machines of type  $k \in \{1, 2, \dots, w\}$ , and let  $w_k = |M_k|$  be the number of identical machines of type  $k$  provided that equality  $|M| = \sum_{k=1}^w w_k$  holds. Competitions among operations that have to be processed by machines of the same type are represented via edge set  $E$ , namely: edge  $[i, j]$  belongs to set  $E$  if and only if operations  $i$  and  $j$  belong to the same set  $V_k$ .

If  $r \in V_k$  and  $s \in V_l$  with  $k \neq l$ , then operations  $r$  and  $s$  have to be processed by different machines (namely, by machines of different types: one machine from set  $M_k$ , and another machine from set  $M_l$ ). For each pair of operations  $i$  and  $j$  with edge  $[i, j] \in E$  there exist three possibilities represented in Fig. 1.

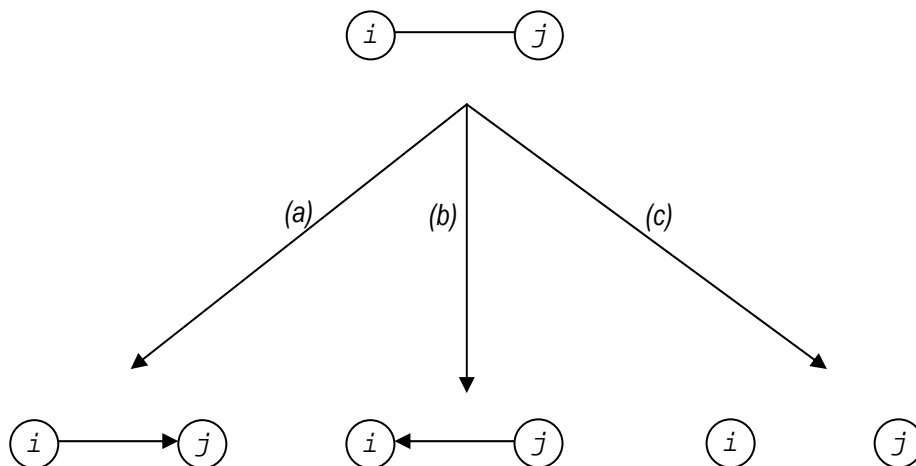


Figure 1. Case (a):  $(i, j) \in A[\alpha^h]$  (

Case (b):  $(i, j) \in A[\alpha^h]$ , respectively) if operations  $i$  and  $j$  are assigned to the same machine, and operation  $i$  (operation  $j$ ) is processed before operation  $j$  (operation  $i$ );

Case (c): edge  $[i, j]$  is removed from the mixed graph  $G$  if operations  $i$  and  $j$  are assigned to two different machines of the same type

Therefore, assigning operations  $V$  to machines  $M$  and sequencing operations assigned to the same machine correspond to orienting or removing edges from set  $E$  in one of these three possibilities. If all such edge transformations do not cause conflicts in the resulting digraph, then they define assigning operations  $V_k$  to machines  $M_k$ ,  $k = 1, 2, \dots, w$ , and define the sequence of operations assigned to each machine  $\mu_u \in M$ . Acyclic digraph constructed in such a way corresponds to a feasible schedule, if the number  $w_k$  of the available machines of each type  $k$  is sufficient for such a transformation of edges in set  $E$ . More precisely, selecting edges  $[i, j]$  one by one from set  $E$  of the mixed graph  $G$ , and applying to each edge  $[i, j]$  one of the three possible transformations (removing edge  $[i, j]$ , or substituting it either by arc  $(i, j)$  or by arc  $(j, i)$ ), we obtain transformation sequence  $\alpha^h = (\alpha^1, \alpha^2, \dots, \alpha^{|E|})$  of all the edges in the mixed graph  $G$ . Let  $G[\alpha^h] = (V, A[\alpha^h], \emptyset)$  denote a digraph obtained from the mixed graph  $G$  as a result of the above transformation  $\alpha^h$ . In article [15], the following claim has been proven.

**Theorem.** *Digraph  $G[\alpha^h]$  defines a feasible schedule if and only if the following three conditions hold:*

- 1) *digraph  $G[\alpha^h]$  contains no circuits;*
- 2) *for any  $k \in \{1, 2, \dots, w\}$ , the number of components in subgraph  $(V_k, A_k[\alpha^h], \emptyset)$  of digraph  $G[\alpha^h]$  is not greater than  $w_k$ ;*
- 3) *for any  $k \in \{1, 2, \dots, w\}$ , each component of the digraph  $(V_k, A_k[\alpha^h], \emptyset)$  is a tournament.*

Let  $P(G)$  denote the set of all digraphs  $G[\alpha^h]$  satisfying all conditions 1), 2) and 3) given in Theorem. To find optimal schedule we may enumerate (e.g., implicitly via B&B method) digraphs from set  $P(G)$ , and choose the *optimal digraph*, i.e., that with minimal value of the given objective function.

Edge  $[i, j] \in E$  is called *conflict* if both its orientations  $(i, j)$  and  $(j, i)$  in the mixed graph  $G$  cause an increase of the starting time of at least one operation from set  $V$ . Using *conflict resolution strategy*, we deal with one conflict edge at each iteration of the B&B algorithm and test three possible transformations of this edge. The rule for choosing the edge from the set of conflict ones may be different, but the common idea is to choose that with maximal value of the conflict measure [4, 14, 15]. To find a solution of the scheduling problem, an *exact* B&B algorithm may be used to realize implicit enumeration of all feasible schedules. An *approximate version* of a B&B algorithm may be obtained from exact one using some error bound allowed for the desired objective function value. A *heuristic*



algorithm may be constructed by considering only one branch of the solution tree via choosing at each iteration single transformation of the conflict edge (e.g., due to some priority rules used for choosing operation from the set of two operations involved in conflict). Thus, mixed graph model with conflict resolution strategy may be a foundation for constructing a variety of scheduling algorithms differing in the running time and in accuracy of the best solution obtained.

In the learning technology, we use the exact (or approximate version of) B&B algorithm in Module 1, while in Module 3 we use adaptive algorithm with logical rules produced within Module 2 by inductive inference. In the following sections, we show how to realize such an adaptation using mixed graph  $G$ .

### Stage of Learning (Module 1 and Module 2)

At the stage of acquiring knowledge, we accumulate the knowledge about scheduling problems under consideration. Module 1 solves a set of sample scheduling problems via exact B&B algorithm (or via approximate B&B algorithm with allowed upper bound of the objective function) and stores information on the successful local decisions that have led to the optimal schedule (or to the best schedule obtained). The main idea for acquiring knowledge is to accumulate (in so-called *learning table*) the information that was obtained during the use of the time-consuming exact (or approximate version of) B&B algorithm. Next, we show how this knowledge could be used when solving other scheduling problems that are close to previous ones already solved by Module 1 (i.e., problems with similar structure of the processing system, close numerical data, the same objective function, etc.).

The aim for constructing learning table is to accumulate the knowledge on successful local decisions used in the historical performance of the B&B algorithm. The stage of learning consists in accumulating information and tuning some parameters of the B&B algorithm to the properties of a set of scheduling problems to be solved. The learning table is analogous to those used in *pattern recognition*. It describes which transformation of a conflict edge seems to be preferable. While solving the sample problem, information on successful transformations of the conflict edges has to be written in the learning table (see Table 1 without last row). Each row of this table corresponds to one conflict edge. After the B&B algorithm from [15] is stopped, the path in the solution tree is known that has led to the optimal schedule (or to the best schedule obtained). We can consider the set of conflict edges tested in this path. If the B&B algorithm chooses the most conflict edge  $[i_1, j_1]$  as the first one, then Module 1 considers edge  $[i_1, j_1]$  as the first one too, and so on. Columns in Table 1 corresponding to characteristics are filled by calculated characteristic values  $x^i$  of the object (conflict edge)  $[i_m, j_m]$ . The last column in Table 1 contains classes  $\Omega_1, \Omega_2, \dots, \Omega_7$  attributed to objects  $[i_m, j_m]$ . Let  $T_{u,v}$  denote the learning table with  $v$  objects and  $u$  characteristics. In our software, number  $u$  is the parameter of the program fixed at the initial step of the algorithm, while number  $v$  of the objects may grow with the growth of the number of sample problems solved by Module 1.

Table 1: The learning (recognition) table  $T_{u,v}$  used in Module 2 (Module 3)

Objects (conflict edges)	Characteristic values				Classes of objects $\Omega_1, \Omega_2, \dots, \Omega_7$
	$x^1$	$x^2$	....	$x^u$	
$[i_1, j_1]$	$b^1$	$b^2$	...	$b^u$	$\Omega_{k1}$
$[i_2, j_2]$	$b^1$	$b^2$	...	$b^u$	$\Omega_{k2}$
...	...	...	...	...	...
$[i_v, j_v]$	$b^1$	$b^2$	...	$b^u$	$\Omega_{kv}$
$[i, j]$	$b^1$	$b^2$	...	$b^u$	?

Notion of characteristic like  $x^i$  is important for object description in any knowledge-based system. Next, we demonstrate how characteristics of the conflict edges may be introduced by generalizing priority rules. While creating the learning table for the first time, the characteristics are chosen from the known list, and they are ordered with respect to their theoretical informative measure. Each characteristic may correspond to a priority rule

for a conflict resolution. After words having adopted on the set of scheduling problems, Modules 1 and 3 may also use more complex and more informative logical rule as a new characteristic created by Module 2. In this case, useless (non-informative) characteristic  $x^u$  has to be rejected from table  $T_{u,v}$  (since number  $u$  of characteristics used in table  $T_{u,v}$  is fixed at the initial step of the algorithm). Index  $m$  from set  $\{1, 2, \dots, u\}$  and a position in the ordered set of characteristics have to be prescribed to the new characteristic with respect to its usefulness: the more informative characteristic is, the smaller index  $m$  it must have. Such a replacement of characteristics is realized within Module 2. In the experiments, we used first the characteristics corresponding to the priority rules given in [9]. A priority rule allows compare priority values of two operations competing on machines of the same type. It is useful to introduce the corresponding characteristic equal to the difference between these priority values for two operations  $i$  and  $j$  for the conflict edge  $[i, j]$ . To make such characteristics more universal (and so applicable to other scheduling problems), it is reasonable to use the relative difference of these priority values obtained after dividing them by a common value (e.g., by the maximal value of this characteristic in the whole mixed graph  $G$ ). Next, we demonstrate typical examples of characteristics used in computational experiments.

*SPT* rule recommends to process first the operation with the "Shorter Processing Time", and then the operation with the greater processing time. So, we can introduce corresponding characteristic as follows:

$$x_1 = (p_i - p_j) / \max\{p_k : k \in V\}.$$

Value  $x_1$  characterizes edge  $[i, j]$  more adequate than the corresponding priority rule: the sign of this value shows which operation  $i$  or  $j$  has the shorter processing time, and absolute value of  $x_1$  makes it possible to compare edges of the mixed graph  $G$  with those of the mixed graphs constructed for other problems.

*FIFO* rule recommends to process first the operation that has the shorter starting time ("First In, First Out"). The starting time  $s_i$  of operation  $i$  can be easily calculated as the maximal weight of the path ending in vertex  $i$  of the digraph  $(V, A, \emptyset)$ . Value  $x_2$  of a characteristic corresponding to priority rule *FIFO* may be calculated for edge  $[i, j]$  as follows:

$$x_2 = (s_i - s_j) / \max\{s_k : k \in V\}.$$

In the computational studies (realized for the special case of the problem with  $|M_k| = 1$  for each  $k = 1, 2, \dots, w$ ) especial attention has been paid to priority rule *LIOF* ("Least Increase in the Objective Function"). Considering two operations  $i$  and  $j$  processed by the same machine, two values  $F_{ij}$  and  $F_{ji}$  of the objective function  $F$  are calculated for two possible orders to process these operations. Value  $F_{ij}$  is calculated for partial schedule with operation  $i$  being processed first and operation  $j$  being processed second (in this case, conflict edge  $[i, j]$  has to be substituted by arc  $(i, j)$ ). Value  $F_{ji}$  is calculated for partial schedule with operation  $j$  being processed first and operation  $i$  being processed second (edge  $[i, j]$  has to be substituted by arc  $(j, i)$ ). The *LIOF* rule recommends to choose the order defined by arc  $(i, j)$  for operations  $i$  and  $j$  if  $F_{ij} < F_{ji}$ , and to choose the opposite order defined by arc  $(j, i)$ , otherwise.

The computational studies with three famous job shop problems from [7] showed that *LIOF* rule is very informative. More precisely, it was shown (by experiments on computer) that while constructing an optimal schedule for the makespan criterion  $C_{max} = \max\{C_i : i \in V\}$  (where  $C_i$  means the completion time of job  $J_i \in J$ ) via Module 1, the most part of the conflict edges was successfully oriented just due to the *LIOF* rule. For test problems, number of edges substituted by arcs with respect to *LIOF* rule formed about 60 percent, and the number of exceptions formed often only 1 percent of the whole number of conflict edges tested in Module 1. For the rest 39 percent of conflict edges  $[i, j]$ , operations  $i$  and  $j$  were often equivalent in the sense of equality  $F_{ij} = F_{ji}$ .

Thus, at the learning stage, we consider the conflict edge of the mixed graph  $G$  as an *object* (using pattern recognition term). All columns in table  $T_{u,v}$  (except the first and the last ones) contain characteristic values of the objects. They form the description-vector for each conflict edge tested while constructing optimal schedule via Module 1. The first column enumerates conflict edges in the order as they have been tested in the B&B algorithm on the path to the optimal schedule in the solution tree (or to the best schedule obtained). The last column in table  $T_{u,v}$  is used to classify objects in order to compare them with new objects within Module 3. At the solution stage (Module 3), the description-vector of the object is used to decide what edge substitution or removal has more probability to be successful.

---

## Classes of Objects

---

To identify what transformation of conflict edge has been successful in the historical performance of the B&B algorithm, we introduce seven classes of objects:  $\Omega_1, \Omega_2, \dots, \Omega_7$  (see last column in Table 1). In what follows, it is assumed that inequality  $i < j$  holds for each edge  $[i, j] \in E$ .

Object  $[i, j]$  belongs to class  $\Omega_1$  if it is preferable to substitute this edge by arc  $(i, j)$  in order to obtain effective schedule with respect to the given objective function. (In such a case, we shall say that arc  $(i, j)$  *dominates* both arc  $(j, i)$  and edge  $[i, j]$ ).

Object  $[i, j]$  belongs to class  $\Omega_2$  if it is preferable to substitute it by arc  $(j, i)$ , and we say that arc  $(j, i)$  dominates both arc  $(i, j)$  and edge  $[i, j]$ .

Object  $[i, j]$  belongs to class  $\Omega_3$  if it is preferable to remove conflict edge  $[i, j]$  from the mixed graph  $G$ , and we say that edge  $[i, j]$  dominates both arcs  $(i, j)$  and  $(j, i)$ .

If there is no enough information to decide what sole edge transformation is preferable, we have to include object  $[i, j]$  to one of the classes  $\Omega_4, \Omega_5, \Omega_6$  or  $\Omega_7$  depending on available information. Namely, if arcs  $(i, j)$  and  $(j, i)$  simultaneously dominate edge  $[i, j]$  but not each other, then  $[i, j] \in \Omega_4$ . If arc  $(i, j)$  and edge  $[i, j]$  simultaneously dominate arc  $(j, i)$  but not each other, then  $[i, j] \in \Omega_5$ . If arc  $(j, i)$  and edge  $[i, j]$  simultaneously dominate arc  $(i, j)$  but not each other, then  $[i, j] \in \Omega_6$ . Finally, if there is no enough information on arc (or edge) domination, then object  $[i, j]$  has to belong to class  $\Omega_7$ .

Next, we render the concrete edge classification rules realized within our Module 2. The learning process is based on the solution tree constructed for the sample problem. Let scheduling problem  $S$  be solved by an exact B&B algorithm within Module 1. Then we obtain optimal digraph  $G[\alpha^*] = (V, A[\alpha^*], \emptyset)$  that defines optimal schedule. Moreover, the path from vertex  $G$  to vertex  $G[\alpha^*]$  in the constructed solution tree is known. Each vertex of this path is defined by a conflict edge  $[i, j]$ . If  $(i, j) \in A[\alpha^*]$ , then  $[i, j] \in \Omega_1$ . If  $(j, i) \in A[\alpha^*]$ , then  $[i, j] \in \Omega_2$ . If  $(i, j) \notin A[\alpha^*]$  and  $(j, i) \notin A[\alpha^*]$ , then  $[i, j] \in \Omega_3$ . Thus, each object obtained due to the solution of the sample problem via exact B&B algorithm belongs to one of the three classes:  $\Omega_1, \Omega_2$ , or  $\Omega_3$ .

If the sample problem is solved only approximately, object classification is based on the relation between lower and upper bounds of the values of the given objective function. Let scheduling problem  $S$  be solved by an approximate B&B algorithm, then the best digraph  $G[\alpha^\circ] = (V, A[\alpha^\circ], \emptyset)$  constructed gives upper bound  $UB$  that is greater than the minimal lower bound calculated for all the pendant vertices of the constructed solution tree. Thus, we obtain path  $\pi(G - G[\alpha^\circ])$  from vertex  $G$  to vertex  $G[\alpha^\circ]$  in the solution tree. Each intermediate vertex  $G'$  in this path defines conflict edge  $[i, j]$  for further branching. Object  $[i, j]$  has to be included in one of the above seven classes. Next, we describe the rule for this classification in detail only for the case when path  $\pi(G - G[\alpha^\circ])$  includes arc  $(a)$  (see Fig. 1). We will indicate this case as follows:  $(a) \in \pi(G - G[\alpha^\circ])$ . In this case, conflict edge  $[i, j]$  in the mixed graph  $G$  is substituted by arc  $(i, j)$ .

Let  $(a) \in \pi(G - G[\alpha^\circ])$ . In the solution tree, three possible transformations have been realized for conflict edge  $[i, j]$  (see Fig. 1). Let  $LB(j, i)$  ( $LB[i, j]$ , respectively) denote the minimal lower bound calculated for all the pendant vertices  $G''$  to which there exists a path  $\pi(G' - G'')$  from vertex  $G'$  starting with arc  $(b)$ :  $(b) \in \pi(G' - G'')$  (starting with arc  $(c)$ :  $(c) \in \pi(G' - G'')$ , respectively).

If  $LB(j, i) \geq UB$  and  $LB[i, j] \geq UB$ , then  $[i, j] \in \Omega_1$ . If  $LB(j, i) < UB$  and  $LB[i, j] \geq UB$ , then  $[i, j] \in \Omega_4$ . If  $LB(j, i) \geq UB$  and  $LB[i, j] < UB$ , then  $[i, j] \in \Omega_5$ .

Classification rules for other two cases  $(b) \in \pi(G - G[\alpha^\circ])$  and  $(c) \in \pi(G - G[\alpha^\circ])$  are analogous. In particular, if  $(b) \in \pi(G - G[\alpha^\circ])$ , then object  $[i, j]$  has to belong to one of the three classes  $\Omega_2, \Omega_4$  or  $\Omega_6$ . If  $(c) \in \pi(G - G[\alpha^\circ])$ , then object  $[i, j]$  has to belong to one of the three classes  $\Omega_3, \Omega_5$  or  $\Omega_6$ .

At the end of the stage of learning, the information accumulated in table  $T_{u,v}$  is used in Module 2 to construct the *recognition table*, which is used at the solution stage within Module 3. A recognition table has the similar form as table  $T_{u,v}$  has, except the last row added to recognition table (see Table 1). (However, we use the same notation  $T_{u,v}$  for recognition table as well.) In order to construct the recognition table a subset of the most informative characteristics is selected and sorted with respect to their informativeness. Some characteristics that turned out to be not essential for sample problems have to be rejected from table  $T_{u,v}$ .

In [13], it is shown how to select the most informative characteristics. Module 2 systematizes the knowledge obtained using the pattern recognition apparatus, and constructs the generalized logical rule for conflict resolution. Such a rule describes successful local decisions and should be used in Module 3 for more efficient solving large-scale scheduling problems via adopting successful local decisions by principle of analogy.

---

### Solution Stage (Module 3)

---

Scheduling algorithm used as Module 3 occupies intermediate position between heuristic and approximate B&B algorithms depending on the number of sample problems solved exactly or approximately at the stage of learning (Module 1 and Module 2). Conflict resolution strategy is also used in Module 3. If Module 3 chooses conflict edge  $[i, j]$ , it has to recognize to which class from set  $\{\Omega_1, \Omega_2, \dots, \Omega_7\}$  edge  $[i, j]$  has to belong. Thus, a recognition problem arises, and it is reasonable to adopt to a new situation the strategy that has led to successes at the learning stage in Module 1.

In order to solve the recognition problem, Module 3 uses the procedure based on the bound calculation. A resemblance bound is calculated that characterizes the distance between the description vector of recognizable object  $[i, j]$ , and that of sample objects with respect to combination  $H$  of characteristics  $x^1, x^2, \dots, x^u$  used in the recognition table  $T_{u,v}$ . We have to determine edges  $[i_k, j_k]$  given in table  $T_{u,v}$  such that the corresponding vectors  $(b^1, b^2, \dots, b^u)$  are the closest ones to vector  $(b^1, b^2, \dots, b^u)$  with respect to the characteristics  $x^1, x^2, \dots, x^u$ . Two objects  $[i, j]$  and  $[i_k, j_k]$  are considered to be similar with respect to the system  $H$  of characteristics, if at least  $\delta$  inequalities  $|b^j - b^k| \leq \varepsilon_k$  are satisfied with  $j = 1, 2, \dots, u$  and  $x^j \in H$ , where  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_u$  and  $\delta$  being parameters of Module 3. Bounds are calculated for the number of objects of the learning table that are similar (close) with respect to system  $H$ . The analysis of such bounds allows to decide to which class  $\Omega_1, \Omega_2, \dots, \Omega_7$  object  $[i, j]$  has to be assigned. Due to this calculation, one can decide what transformations are preferable for the conflict edge  $[i, j]$ , and how many transformations (one, two or three) of this edge have to be treated in Module 3. If  $[i, j] \in \Omega_1$ , then solution tree has to contain arc (a) (see Fig. 1). If  $[i, j] \in \Omega_2$ , then solution tree has to contain arc (b). If  $[i, j] \in \Omega_3$ , then solution tree has to contain arc (c). If  $[i, j] \in \Omega_4$ , then solution tree contains two arcs (a) and (b). If  $[i, j] \in \Omega_5$ , then solution tree contains two arcs (a) and (c). If  $[i, j] \in \Omega_6$ , then solution tree contains two arcs (b) and (c). If  $[i, j] \in \Omega_7$ , then solution tree contains all three arcs (a), (b) and (c) that are represented in Fig. 1.

Due to this approach, we obtain close interaction between implicit enumeration algorithm (Module 1), approximate algorithm and heuristics (in Module 3). Effective priority rules obtained via Module 2 contribute to the implementation of the effective upper and lower bounds in the branch-and-bound algorithm (Module 1). Suitable heuristics have to be incorporated in the branch-and-bound procedures (Module 1 and Module 3) to compute the solutions corresponding to the most promising part of the solution tree.

---

### Conclusion

---

The use of an exact scheduling algorithm is computationally expensive and so impracticable. At present, rather common methodology carried out by practitioners is the use of simple heuristics based often on weakly grounded dispatching rules which do not ensure desirable accuracy of the obtained solution. To overcome these difficulties, we propose a knowledge-based system that may be used for the adaptation of the approximate version of the B&B algorithm for the specific practical scheduling problems. The solution process is partitioned into two stages: the *learning* stage realized by Module 1 and Module 2 for small (sample) scheduling problems, and the *solution* stage realized by Module 3 for moderate and large (real-world) scheduling problems.

Module 1 creates the database of learning information for the certain class of the scheduling problems and enlarges it while solutions of new problems become available. At the stage of learning, one can also use practically effective schedules obtained via commercial packages such as that described in [5]. The volume of information, its correctness and absence of noises depends on the quality of the scheduling algorithms applied in the framework of Module 1 and on the computing time given for learning process (Module 1 and Module 2). More exact algorithms requiring larger computing time may ensure more informative knowledge. Therefore, it is important to use effective algorithms within Module 1. The algorithm used in Module 3 should consider at each step the same question as the algorithm used in Module 1. The bridge linking Module 1 and Module 3 is

Module 2, which automates the learning of peculiarities of the similar scheduling problems and may give foundation for creating generalized logical rule for proper conflict resolution. As a result, Module 2 generates suitable priority rules, which may be considered in Module 3 as new comprehensive heuristics developed just for the practical scheduling problems under consideration.

This research was partially supported by INTAS (project 03-51-5501), and by the Belarusian Republican Foundation for Fundamental Research.

---

## Bibliography

---

- [1] Chan F.T.S., Chan H.K. Dynamic scheduling for a flexible manufacturing system – the pre-emptive approach. *International Journal of Advanced Manufacturing Technology*, 2001, 17 (10), 760–768.
- [2] Dorndorf U., Pesch E. Evolution based learning in a job shop scheduling environment. *Computers & Operations Research*, 1991, 22 (1), 25–40.
- [3] Hertz A., Widmer M. Guidelines for the use of meta-heuristics in combinatorial optimization. *European Journal of Operational Research*, 2003, 151, 247–252.
- [4] Jain A.S., Meeran S. Deterministic job-shop scheduling: past, present and future. *European Journal of Operational Research*, 1999, 113, 390–434.
- [5] Janicke W. Excel planning interface for the SAP R/3 system. *Chemical Engineering & Technology*, 2001, 24 (9), 903–908.
- [6] Kim M.H., Kim Y-D. Simulation-based real-time scheduling in a flexible manufacturing system. *Journal of Manufacturing Systems*, 1994, 13 (2), 85–93.
- [7] Muth J.F., Thompson G.L. *Industrial scheduling*. Englewood Cliffs, N.Y., Prentice Hall, USA. 1963.
- [8] Nof S.Y., Grant F.H. Adaptive/predictive scheduling: review and a general framework. *Production Planning and Control*, 1991, 2 (4), 298–312.
- [9] Panwalkar S.S., Iskander W. A survey of scheduling rules. *Operations Research*, 1977, 25, 45–61.
- [10] Paternina-Arboleda C.D., Das T.K. A multi-agent reinforcement learning approach to obtaining dynamic control policies for stochastic lot scheduling problem. *Simulation Modelling Practice and Theory* (to appear, available online).
- [11] Priore P., De La Fuente D., Gomez A., Puente J. A review of machine learning in dynamic scheduling of flexible manufacturing systems. *AI EDAM*, 2001, 15 (30), 251–263.
- [12] Priore P., Fuente D.L., Gomez A., Puente J. Dynamic scheduling of manufacturing systems with machine learning. *International Journal of Foundation of Computer Science*, 2001, 12 (6), 751–762.
- [13] Shakhlevich N.V., Sotskov Yu.N., Werner F. Adaptive scheduling algorithm based on mixed graph model. *IEE PROCEEDINGS – Control Theory and Applications*, 1996, 43 (1), 9–16.
- [14] Sotskov Yu.N. Software for production scheduling based on the mixed (multi)graph approach. *Computing & Control Engineering Journal*, 1996, 7 (5), 240–246.
- [15] Sotskov Yu.N. Mixed multigraph approach to scheduling jobs on machines of different types. *Optimization*, 1997, 42, 245–280.
- [16] Tanaev V.S., Sotskov Yu.N., Strusevich V.A. *Scheduling theory. Multi-stage systems*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.
- [17] Wang J.M., Usher J.M. Application of reinforcement learning for agent-based production scheduling. *Engineering Applications of Artificial Intelligence*, 2005, 18 (1), 73–82.
- [18] Whiteson S., Stone P. Adaptive job routing and scheduling. *Engineering Applications of Artificial Intelligence*, 2004, 17 (7), 855–869.
- [19] Wiers V.C.S. A review of the applicability of OR and AI scheduling techniques in practice. *Omega – International Journal of Management Science*, 1997, 25 (2), 145–153.

---

## Authors' Information

---

Yuri N. Sotskov – United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Surganova str., 6, Minsk, Belarus; e-mail: [sotskov@newman.bas-net.by](mailto:sotskov@newman.bas-net.by)

Nadezhda Sotskova – Hochschule Magdeburg-Stendal, Fachbereich Wasserwirtschaft, PSF 3680, 39011 Magdeburg, Germany; e-mail: [drsotskova@hotmail.com](mailto:drsotskova@hotmail.com)

Leonid V. Rudoi – United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Surganova str., 6, Minsk, Belarus; e-mail: [marqol@newman.bas-net.by](mailto:marqol@newman.bas-net.by)

## ROBOT CONTROL USING INDUCTIVE, DEDUCTIVE AND CASE BASED REASONING

Agris Nikitenko

*Abstract:* The paper deals with a problem of intelligent system's design for complex environments. There is discussed a possibility to integrate several technologies into one basic structure that could form a kernel of an autonomous intelligent robotic system. One alternative structure is proposed in order to form a basis of an intelligent system that would be able to operate in complex environments.

The proposed structure is very flexible because of features that allow adapting via learning and adjustment of the used knowledge. Therefore, the proposed structure may be used in environments with stochastic features such as hardly predictable events or elements. The basic elements of the proposed structure have found their implementation in software system and experimental robotic system. The software system as well as the robotic system has been used for experimentation in order to validate the proposed structure - its functionality, flexibility and reliability. Both of them are presented in the paper. The basic features of each system are presented as well. The most important results of experiments are outlined and discussed at the end of the paper. Some possible directions of further research are also sketched at the end of the paper.

*Keywords:* Artificial intelligence, Inductive reasoning, Deductive reasoning, Case based reasoning, Machine learning, Learning algorithms.

*ACM Classification Keywords:* I.2.9 Robotics; I.2.3 Deduction and Theorem Proving: Deduction, Induction, Uncertainty, Fuzzy, and Probabilistic Reasoning; I.2.6 Learning

---

### Introduction

---

During a short period of time (lasting only several decades) there have been developed a lot of different technologies and approaches to solve various types of problems existing in the field of artificial intelligence. A complexity of those tasks that can be performed by intelligent systems is growing from year to year. Therefore, the range of application of artificial intelligence (AI) technologies has been significantly widened. One of the challenging tasks that is always hard to accomplish is building an autonomous intelligent robotic system, because it includes design of software, hardware and mechanics as well. The task is even more challenging if an operation environment is complex and has some stochastic features or entities with stochastic behavior.

Before trying to build a structure of an autonomous intelligent robotic system, it is necessary to define the environment in which the system should be able to operate. A basis of such a definition can be found in the assumption that every object can be described as a system [Lit.1] Obviously, a complex environment can be described as a complex system. There are several very basic features that define a complex system [Lit. 2]:

- uniqueness – usually complex systems are unique or number of similar systems is unweighted.
- hardly predictable – complex systems are very hard to predict. It means that it is hard to calculate the next state of a complex system if the previous states are known. The hard predictability may be related of the mentioned stochastic elements or features of the environment.
- an ability to maintain some progress resisting against some outer influence (including influence of the intelligent system).

Of course, any complex system has every general feature such as a set of elements, a set of relations or links etc... [Lit. 3]

Obviously, if the system operates autonomously in a complex environment, it has to form some model of the environment. It is not always possible to build a complete model of the environment for different reasons.

That may be caused by a huge space of possible states of the environment (or even infinite), expanses or other reasons. It means that an intelligent system will use only an incomplete model of the environment during operation and be able to achieve its goals.

The structure presented in the paper exploits an adaptation and uncertain reasoning technique as general methods to deal with an incompleteness of the system's model of its environment.

---

### Basic Feature of the Intelligent System

---

In this section the basic features of the proposed structure are outlined and explained according to the previous research activities [Lit.4].

Summarizing the basic features of the proposed structure are as following:

- An ability to generate a new knowledge from the already existing in the system's knowledge base. This ability can be achieved by means of deductive reasoning. In order to increase the efficiency a case based reasoning may be used [Lit.6] This feature, obviously, includes also an ability to reason logically. The proposed structure does not state the kind of deductive reasoning that should be used. The only rule is that the selected deductive reasoning method has to address demands of a particular task. As it is described above, complex environments may be very dynamic and even with stochastic features. Therefore, some uncertain reasoning techniques may be the most suitable for complex environments. The described below experimental systems also have uncertain reasoning techniques implemented as a deductive reasoning module.
- An ability to learn. As it is assumed above in complex environments the intelligent system eventually will not have a complete model of the environment. Therefore, the environment will be hardly predictable. Also complex environments are dynamic - in other words the system will face with new situations very often. Obviously, some mechanisms of adaptation should be utilized. From point of view of intelligence, an adaptation includes the following main capabilities: a capability of acquiring new knowledge and adjustment of the existing knowledge. The inductive reasoning module refers to capability of acquiring new knowledge or learning, in other words. This feature may be implemented by means of inductive reasoning. During an operation, the intelligent system collects a set of facts through sensing the environment that forms input for learning.
- An ability to reason associatively. This feature is necessary due to a huge set of possible different situations that the intelligent system may face with. For example, there may be two different situations that can be described by  $n$  parameters ( $n$  is big enough number) where only  $k$  parameters are different ( $k$  is small enough number). Obviously, these situations may be assumed as similar. Therefore an associative reasoning is used – to reason about objects or situations that are observed for the first time by the intelligent system similarly to reasoning about known situations and using knowledge about the known situations and object. The associative reasoning is realized through using associative links among similar objects and situations. An issue about which objects and situations should be linked is conditioned by particular tasks or goals of the system's designer.
- An ability to sense an environment. This feature is essential for any intelligent system that is built to be more or less autonomous. This feature also includes an ability to recognize objects / situations that the system has faced with as well as an ability to obtain data about new objects. All sensed data is structured in frames (see below). During the frame formation process the sensed environment's state is combined with system's inner state thereby allowing the system to reason about system itself as well as relation between system's inner state and sensed environment's state. Also, the sensed system's and environment's states are used to realize a feedback in order to adjust system's knowledge. Thus, the system's flexibility is increased.
- An ability to act. This feature is essential for any intelligent system that is designed to do something. If the system (autonomous) is unable to act, it will not be able to achieve its goals. Obviously, the system has to act in order to achieve its goals as well as to obtain the feedback information for readjusting its knowledge or to learn new knowledge. The way of acting and the purpose of acting vary depending of goals of the system's designer or user.

The listed above features form the basis for an intelligent system that operates in sophisticated environment. According to the features of complex systems that are listed above, any of them may be implemented, as it is needed for a particular task. In other words, the implementation methods and approaches are dependent of the purposes of the system itself.

Nevertheless the main question is how to bind all of the listed above features in one whole - one intelligent system.

Obviously, there is a necessity for some kind of integration. There are many good examples of different kinds of integration. For example so called soft computing which combines fuzzy logic with artificial neuron nets [Lit.6] or Case based reasoning combined with deductive reasoning [Lit.7].

In order to adjust an intelligent system for some particular tasks different structures may be used [Lit 14]. This paper presents one of the alternative structures that may be used in order to implement all of the listed above features and may form a kernel of autonomous intelligent system.

The proposed structure is based on intercommunicating architecture. In other words, the integrated modules are independent, self-contained, intelligent processing modules that exchange information and perform separate functions to generate solutions [Lit. 14].

### Structure of the Intelligent System

According to the list of very basic features there can be outlined the basic modules that correspond to the related reasoning techniques.

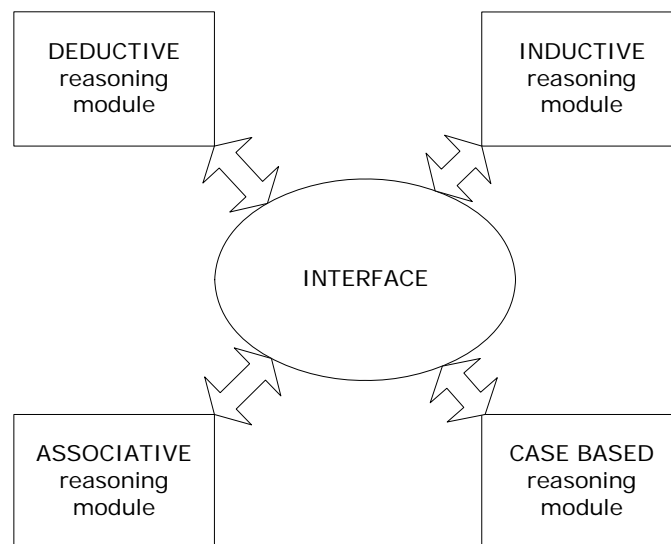


Figure 1. Basic modules

As it is outlined in the figure 1, there are four basic modules that form system's kernel. Each of the modules has the following basic functions:

– *Deductive reasoning module*

This module performs deductive reasoning using *if..then* style rules. In order to implement adaptation functionality, this module may exploit some of uncertain reasoning techniques. In the proposed structure the main purpose of this module is to predict future states of the environment as well as the inner state of the system. During the reasoning process *if..then* rules are used in combination with input data obtained from system's sensors. The proposed structure itself does not state what kind of deductive reasoning method should be used. It depends of particular goals of the system's designer. In both practical implementations a forward chaining certainty factor based reasoning had been used [Lit.17]. If a task requires fuzzy reasoning or other reasoning technique may be used as well.



– *Inductive reasoning module*

This module performs an inductive reasoning or in other words – inductive learning. It learns new rules and adds them to the rule base. Also, the incoming data from system's sensors is used. Again, the proposed structure does not state what kind of inductive learning technique is used. The only limitation is a requirement to produce rules that could be used by the deductive reasoning module. For example, if the fuzzy reasoning is used, then the result has to include fuzzy rules.

– *Case based reasoning module*

Case based reasoning operates with "best practice" information that helps to reduce planning time as well as provides this information for researcher or user in explicit manner. As said above in complex environments there may be a while of unique situations. To extract (or to learn) any rule an intelligent system needs at least two equal (or similar – the most part of feature (attributes) are equal) situations. It means that in complex environments a while of situations experienced by the intelligent system may remain unused. Obviously, these unique situations (or cases) may be extremely valuable not only for the intelligent system but also for the modeler who uses the system. The case based reasoning module is involved to process and use these unique situations.

– *Associative reasoning module*

This module links objects according to similarities among object features as well as situations, thus allowing to reason associatively [Lit.5]. This module allows to reason about new situations or new objects using knowledge about similar objects or situations. It is an essential ability in complex and dynamic environments in order to increase a flexibility of the intelligent system.

Of course, the intelligent system needs additional modules that would supply it with necessary information about the environment and mechanisms to perform some actions.

Therefore, the basic structure shown in figure 1 is complemented with additional modules. The enhanced structure is depicted in the figure 2.

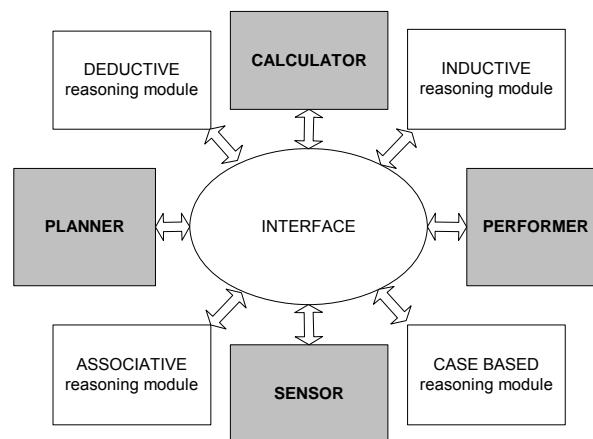


Figure 2 Enhanced structure

The additional modules are drawn in gray. Each of the additional modules has the following basic functions:

- *Planner module*. This module is one of the central elements of the system. Its main function is to plan future actions that lead to achievement of goals of the system. On author's opinion, an ability to predict future events or situations in the most obvious manner demonstrates an intelligence of any more or less autonomous intelligent system. During the planning process three of the basic reasoning techniques are involved – deductive, case based and associative reasoning. A result of the planner is sequence of actions that are expected to be accomplished by the system thereby achieving its goals.
- *Sensor module*. The module's purpose is to collect information from system's sensors about environment and system's inner state. The sensed information is portioned in separate frames (see below) and forwarded

to the interface (discussed later). Once the information is forwarded, it is available for other modules for readjustment of knowledge, for learning new knowledge or other purposes.

- *Performer module.* This module performs a sequence of actions that are listed in the plan. Also, this module uses information about current state of the system an environment in order to determine whether the instant actions can be accomplished. If not appropriate feedback information is sent to the sensor module.
- *Calculator module.* This module collects and produces any reasoning relevant quantitative data. For example, in both implementations (see below) this module is used to calculate certainties of rules including rules newly generated by the inductive reasoning module. Thus, this module is directly involved in knowledge readjustment process. Functionality of the module may be enhanced according to the necessities of the particular tasks or goals of the system's designer.

As it is depicted in figure 1, all of the four modules need some interface to communicate with each other therefore all of the modules use a central element – *Interface* in order to communicate to each other. They are not communicating to each other directly thereby a number of communication links is reduced as well as all of information circulating in the system is available for any module, if there is such a necessity.

A simplified structure of the interface is depicted in the following figure:

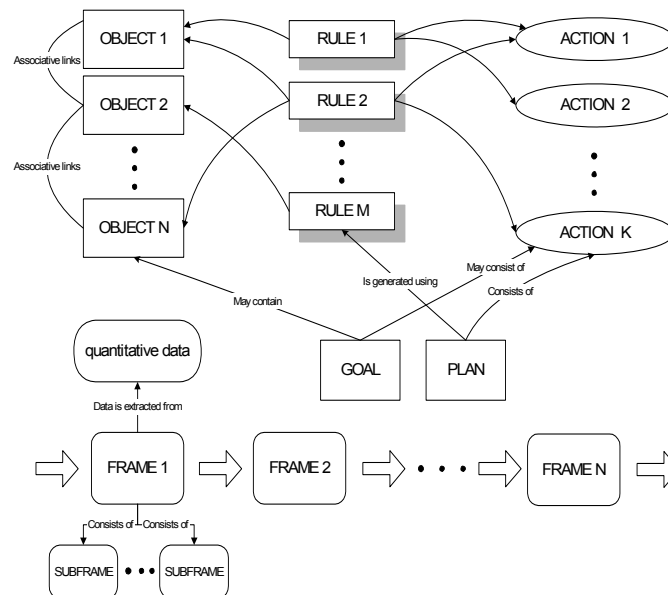


Figure 3 Structure of the interface.

The structure consists of several basic elements. The fundamental element of whole structure is *object*.

**Object.** Objects are key elements in the interface structure. They correspond to some kind of entities in the environment (or in the intelligent system). Every object is described with a set of features (attributes). Each feature has some value. As it is depicted in the figure 3 objects are linked to each other by associative links. These links form basis for associative reasoning.

When the intelligent system runs into a new situation some subset of objects is activated. These objects map to those entities that the intelligent system currently senses. If there is no rule that can be activated, then the intelligent system may try to activate associated objects. Thus, the system can try to reason about objects by using associated rules. The result may be less feasible, but using association among objects the system can run out of dead end situations.

A mechanism of associative memory is very useful when the system works with a noisy data. This mechanism allows to correct faults of the sensing mechanism [Lit.5,Lit.7]. For example, if the input vector of the sense, which corresponds to some entity, has some uncertain or incorrect elements (attributes of object) then the system would

not be able to activate any of the objects. In this case associative memory mechanism will activate the closest object [Lit.5, Lit.7] thus the sensing error will be less significant to the reasoning process.

**Rules.** Rules are any kind of notation that represents causalities. In the practical experiments were used a well-known *if..then* notation. As it is depicted in the figure 3 rules are linked to objects and actions. When the system activates objects by using associative links linked rules also are activated thus system can scan a set of "associated" rules as well. This ability improves system's ability to adapt. Rules (for example, those of type *If..Then*) may refer not only to facts but also to actions. Thereby rules through deductive reasoning are used in planning process.

**Actions.** Actions are some kind of symbolic representation that can be translated by the intelligent system and cause the system to do something. For example "*turn to the right*" causes the system to turn to the right by 90°. Each action consists of three parts: precondition, body and postcondition. Precondition is every factor that should be true before the action is executed. For example, before opening the door it has to be unlocked. Body is a sequence of basic (or lower level) actions that are executed directly – for example a binary code that forwarded to motor controller causes the motors to turn (in a case of robotic system). Post conditions are factors that will be true after the execution of action. For example after opening the door, the door will be opened. It is important to stress that both implementations of the structure do not have any postcondition information at the beginning. All of the postconditions are learned during the system's runtime.

**Frames.** Frames are some kind of data structures that contain the sense array from environment and from the system. It means that frames contain snapshots of the environment's and the system's states.

As it is depicted in figure 3 frames are chained one after another thus forming a historical sequence of the environment's and the system's states. Frames can be structured in hierarchies. Hierarchies help to see values of features that can't be seen in a single snapshot. For example: motion trajectories of some object etc. Frames form an input data for learning (induction module) algorithms as well.

**Goal.** A goal is some kind of task that has to be accomplished by the system. It can be defined in three ways: as a sequence of actions that should be done, as some particular state that should be achieved or as a combination of actions and states. The third option is implemented in robotic systems described below.

**Plan.** Plan is a sequence of actions that is currently executed by the system (Performer module). It may be formed using both basic and complex actions. After the plan is accomplished, it is evaluated depending on whether the goal is achieved or not thereby forming a feedback information for calculator module.

**Quantitative data.** This element is used to maintain any kind of quantitative data that is produced by calculator module and is used during the reasoning process. For example, it may contain certainties about facts or rules, possibilities etc. Quantitative data is collected during the reasoning process as well as during the analysis of the input data - feedback data.

All of those components together form an interface for the basic modules: Inductive, Deductive, Case based and Associative reasoning.

Fundamental elements of the structure are implemented in experimental software and robotic systems that are shortly described below.

---

## Experimental Software System

---

As it is mentioned above, fundamental elements of the proposed structure have found their implementation in experimental software systems. The implemented elements are: Case based reasoning, Inductive reasoning and Deductive reasoning. Deductive reasoning is implemented as a statement logic module based on rules designed in *if...then* manner. The induction module is implemented using very well known algorithm ID3 [Lit.9] It has its more effective successor C4.5 [Lit.10]. The case based reasoning module is implemented using pairs {situation, action}. Each of pair has its value that determines how effective it is in a particular case. During the planning this value determines which actions are selected if more then one action may be selected. The maximum length of the plan is limited in order to avoid infinite planning due to lack of a necessary knowledge for successive planning.

The environment is implemented as world of rabbits and wolf (domain of prey and hunter). There are defined additional objects "obstacles". The number of rabbits and obstacles is not specified thus allowing definition of very complex configurations of the environment. The intelligent system is implemented as wolf. Rabbits may be moving or standing at one place. Wolf can catch rabbits. The wolf is moving according to its plan. Researcher (modeler) can freely change number and place of obstacles and rabbits during the system's runtime thus acting as a stochastic element in the system's environment. The goal also may be defined and changed at any time by the researcher (modeler).

The intelligent system demonstrates flexibility of the proposed structure. The results of experiments and experience accumulated during the implementation shows that new types of objects can be introduced without changing the proposed structure.

It means that even being incomplete this structure demonstrates good ability to adapt and to operate

---

## Experimental Robotic System

---

The implemented robotic system is the next step of validation of the proposed structure.

Robotic system is a semi autonomous intelligent system that encapsulates all of the mentioned above elements of the proposed structure and interface among basic modules described above.

### *Mechanics*

The system is a two-wheeled semi-autonomous robot. It has two driving or casting wheels and two auxiliary wheels for balancing. Casting wheels are driven by direct current motors equipped with planetary reducers.

The robot is equipped with the following sensors:

- Eight IR (infrared) range measuring sensors;
- Electronic compass;
- Four bump sensors (two front and two rear micro switches)
- Four driving wheel movement measuring resistors (two for each driving wheel in order to achieve reliable enough measurements).

Two Basix-X [Lit 15] microprocessors are used in order to communicate with PC and perform input data preprocessing and formatting. Processors communicate with each other via RS-232 connection.

Prepared and formatted data as frames (see above) are sent to PC also via RS-232 connection. All other modules of the intelligent system are implemented as PC-based software that has user-friendly interface allowing simple following the system's operation, collection of research sensitive data, changing system's goals etc....

### *Input*

All sensed data is portioned in frames – one frame per second that is sent to PC via RS-232. In order to increase reliability the sensed data is preprocessed. The most important preprocessing is performed with IR ranger data because of the signal errors that should be eliminated. The IR sensors form so-called *situation* that consists of eight integer values. Each of the value describes distance to the closest object at one of eight directions.

Situation data is combined with other sensor data thereby forming an input data array that has the following structure:

[situation, compass, wheel 1 current position, wheel 2 current position,  
wheel 1 last position, wheel 2 last position, action, Bumper 1, bumper 2],

where:

Situation – eight integer values of ranges at appropriate directions;

Compass – System's orientation angel according to the Earth magnetic pole;

Wheel 1/2 current position – integer values that describes current position of the casting wheel

Wheel 1/2 last position – integer value that describes the last position of the casting wheel;

Bumper 1/2 – Boolean values that indicates whether an appropriate bumper switch is on (the robot has run onto obstacle);

Action – integer value that indicates the last action that has been performed;

The arrays are sent to PC each second. PC software stores incoming data into frames – one data array per frame.

#### *Used basic algorithms*

##### *Inductive reasoning module:*

This module is implemented using Quinlan's C4.5 inductive learning algorithm. One of the C4.5 features it that it requires a portion of data with static structure therefore as an input data a portion of frames is used. Each frame has a static structure. A sequence of frames is reformatted in the way that in one input data set record includes two frames – cause frame and consequence frame. For example:

Frame 1; Frame 2

Frame 2; Frame 3:

.....

Frame n; Frame n+1;

Frame n+1; Frame n+2;

Thus pairs of causes and consequences are formed therefore rules of type *If Cause Then Consequence* may be produced.

Obviously, the produced rules can describe events that follow right one after another. This is very important limitation of the proposed technical solution. Therefore, this issue is one of the future research directions in order to enhance the system with ability to reason about longer periods.

##### *Deductive reasoning module:*

This module is implemented via using forward chaining certainty factor reasoning technique. Rules are described in *if...then* manner. Each rule is evaluated with certainty factor that is obtained each time when new input action is accomplished by the robot and appropriate input frame is received. From the newly added frame the feedback information is obtained and certainty values of appropriate rules (that where used in order to plan the accomplished action during the planning process) are recalculated. Thus, the knowledge of the system is adjusted – via feedback information.

After induction when new rules are added, their certainty values are set to the maximum thus indicating that new knowledge is more valuable than the existing. If the new rule is conflicting with any existing rule, than certainty values of the existing rules are decreased, thus priority of the new rule during the deductive reasoning process is increased. This simple mechanism adds a capability to readjust knowledge according to the changing environment.

##### *Associative reasoning module:*

Associative reasoning module uses links among situations (each unique situation is stored in the interface data structure). If in the newly added frame a unique situation appears, than it is stored in the interface data structure. The new situation is linked to the similar situations. Similarities are determined by empirical algorithm.

During the deductive reasoning process, those links are used to select rules about similar situations if none of them is found about the actual situation. Thus, the system can reason associatively about certain situations if there is such a necessity.

##### *Case-based reasoning module:*

This module is used during the planning process. Cases are added after the feedback information about accomplished task (achieved goal) is received. Each case consists of the following parts: Goal, Situation and Plan. In other words, each case indicates what plan may be used in order to achieve certain goal in a certain situation.

*Planner module:*

This module combines deductive, case-based and associative modules in order to plan future actions that lead to achievement of the goal.

As it is possible to define a combined goal consisting of actions, and states sequenced one after another thus forming a sequence of subgoals, for each subgoal a subplan is produced. Planner can produce a plan only for a single goal or subgoal. A special algorithm is built that follows the plan execution during the runtime.

*System's features*

The PC-based software system implements and demonstrates all of the structure's elements mentioned above. The system is built for research purposes only in other words it is built for experiments in order to examine and validate the proposed structure. Therefore the system's user interface is built to be as flexible as possible allowing its user to manipulate with the robot's state, goals and results at the runtime.

The most important features of the system are:

- Ability to work with multiple goals with mixed structure that may include – actions, states or both;
- Ability to adapt via using inductive learning algorithm C 4.5[Lit.10].
- Case-Based reasoning is used to store information about best-practice cases and to use this information during the planning process.
- Ability to reason via using Certainty theory ideas thus allowing addition of new rules that may be conflicting with existing rules in the rule base.
- Ability to reason using associative links among objects (situations).
- The system's knowledge and system's state relevant data is stored and processed in explicit and easy to follow manner thus demonstrating advantages of the used knowledge based techniques.

It is important to stress that at the very beginning of the system's operation it has no information about consequences of each action – it needs to learn them. But if it is necessary the system's rule base may be filled with rules, cases and other research relevant information thus allowing to model some particular state of the system.

All of the necessary experiments are not finished yet - the system is under research process, but even the first experiments demonstrate very good ability of adaptation and learning new sequences of actions in order to achieve goals. All of the conceived experiments may be split in two major groups – experiments with goals that require matching of one action to one goal and goals that require more than one action in order to achieve the goal. Till now only the first group of experiments has been accomplished.

---

**Possible Advances and Future Research**

---

In order to queue actions one after another thereby building a sequence of actions that lead to achievement of the goal a planning module is used. The planning module is built as a classic single goal planner. If there is more than one goal, then the planner builds plans one by one for each goal. Usually for autonomous systems, there is a necessity to work with more than one goal at the same time, for instance, to follow the charge of batteries and to avoid obstacles. If the system is a team member, then the team's goals should be taken on account as well.

In a common situation, the avoidance of obstacles may have a higher priority than following the charge of batteries. If the battery charge is low then global priorities may change. In other words the system should be able to handle so-called global dynamics [Lit. 16] of the plans and their priorities. The mentioned ability is essential in such domains as robot soccer game or other similar very dynamic environments and complex.

The proposed structure cannot handle globally dynamic planning yet. This is one of directions for future research activities.

Obviously, may be there are tasks that cannot be accomplished using a single intelligent system. For example, simulation of some complex environments such as battlefields, transport systems etc. Therefore, there should be used more than one system, thus forming a multiagent environment.

There are different ways to design multiagent system [Lit.11, Lit 12]. In different domains, different solutions may be applied. Referencing to the said above, there may be outlined another direction of farther research and experiments – adjustments of the proposed structure in order to allow the intelligent system operate in a multiagent environment.

One of the most sophisticated problems in such a multiagent environment is communication because every communication parameter may be variable. It is easy to imagine that two intelligent systems may try to communicate using different knowledge representation schemas, different knowledge, different communication protocols, different type of “conversation” (for example: questioning, answering, argumentation etc.) or even different physical communication channels (radio frequency, verbal communication etc.) [Lit.13].

Another important direction of future research activities is adjusting the basic reasoning modules to allow reasoning about longer periods. That would allow planning actions for longer periods of time and to reason about events that could be caused by the system and would take a place after longer period as well.

---

## Conclusions

---

Practical experiments show that the proposed structure may be very flexible even in very changing environments with variable goals. In both cases an adaptation and ability to learn is essential and both of them are persistent in the proposed structure. The proposed structure demonstrates capability to operate autonomously that makes it useful for autonomous robot control purposes.

In spite of the first results that are quite promising there are still some open questions that should be answered in the farther research activities.

---

## References

---

- [Lit.1] В.Н. Волкова «Теория систем и методы системного анализа в управлении и связи» Радио и Связь 1983.
- [Lit.2] В.В. Дружинин, Д.С. Конторов «Системотехника» Радио и Связь 1985.
- [Lit.3] А.Д.Цвиркун «Структура сложных систем» Советское радио 1975.
- [Lit.4] A.Nikitenko, J.Grundspenkis “The kernel of hybrid intelligent system based on inductive, deductive and case based reasoning” KDS2001 conference proceedings, St. Petersburg 2001.
- [Lit.5] Bradley J. Rhodes “Margin Notes Building a Contextually Aware Associative Memory”, *The Proceedings of the International Conference on Intelligent User Interfaces (IUI '00)*, New Orleans, LA, January 9-12, 2000.
- [Lit.6] M. Pickering “The Soft Option” 1999.A.R.Golding,
- [Lit.7] P.S.Rosenblum “Improving Accuracy by Combining Rule-based and Case based Reasoning” 1996.
- [Lit.8] A.M.Wichert “Associative Computation” der Fakultät der Informatik der Univesität Ulm Wrotslav 2000.
- [Lit.9] J.R. Quinlan “Comparing Connectionist and Symbolic Learning Methods” Basser Department of Computer Science University of Sydney, 1990.
- [Lit.10] J.R. Quinlan “Improved Use of Continuous Attributes in C4.5” *Journal of Artificial Intelligence Research* 1996.
- [Lit.11] P.Scerri, D.Pynadath, M.Tambe “Adjustable Autonomy in Real-world Multi-Agent Environments” Information Institute and Computer Science Department University of Southern California, 2000
- [Lit.12] P.Stone, M.Veloso “Multiagent systems: A Survey from a Machine Learning Perspective.” *IEEE Transactions on Knowledge and Data Engineering*. 1996.
- [Lit.13] J.Dale, E.Mamdani “Open Standards for Interoperating Agent-Based Systems” 2000 (last changes)
- [Lit.14] S. Goonatilake, S. Khebbal “Intelligent Hybrid systems” 2002 john wiley & sons
- [Lit.15] © NetMedia Inc. [www.basicx.com](http://www.basicx.com)
- [Lit.16] H. Burkhard, J. Bach, R. Berger, B. Brunswieck, M. Gollin “Mental models for Robot Control” “Advances in Plan – based control of Robotic agents” International seminar proceedings, Dagstuhl Castle, 2001
- [Lit.17] J.J.Mahoney, R.J.Mooney “Comparing methods for refining certainty-factor rule-bases” *Proceedings of 11<sup>th</sup> international workshop on Machine Learning*, Rutgers, 1994.

---

## Author's Information

---

Agris Nikitenko – Doctoral student, Riga Technical University, Department of System theory, Meza street 1 k.4, Riga LV-1048, Latvia, E-mail: [agris.nikitenko@rembox.lv](mailto:agris.nikitenko@rembox.lv)

## TABLES OF CONTENTS OF IJ ITA – VOLUME 12

### Table of Contents of IJ ITA Volume 12, Number 1

Preface .....	3
<i>Luis Fernando de Mingo López, Eugenio Santos Menéndez, Francisco Gisbert</i> Filtered Networks of Evolutionary Processors .....	5
<i>Milena Dobрева, Nikola Ikonov</i> Digitisation of Cultural Heritage: between EU Priorities and Bulgarian Realities .....	12
<i>Charles Newton Price, Renato J. de Sobral Cintra, David T. Westwick, Martin Mintchev</i> Classification of Biomedical Signals using the Dynamics of the False Nearest Neighbours (DFNN) Algorithm ....	18
<i>Alexander Fish, Orly Yacid-Pecht</i> Bottleneck Problem Solution using Biological Models of Attention in High Resolution Tracking Sensors .....	29
<i>Dimitrina Polimirova-Nickolova, Eugene Nickolov</i> Examination of Passwords' Inflation on the Compressing Process of Non-Encrypted Objects .....	35
<i>Zinoviy Rabynovych</i> Conceptual Idea of Identification of Patterns and Problem Solving in the Human's Memory and the Possibilities to Use it in Artificial Intelligent .....	41
<i>Yuriy Belov, Sergiy Tkachuk, Roman Iamborak</i> Model Construction and Research using System Compositional Approach on Natural Hierarchical Neural Networks. Development of Computer Toolbox .....	49
<i>Yuriy Belov, Sergiy Tkachuk, Roman Iamborak</i> Exploration by Means of Computer Simulation of Nonlinear Hierarchical Structure of Neuronal Memory on the Model of Olfactory Bulb .....	57
<i>Valeriy Koval, Sergey Ryabchun, Volodymyr Savyak, Ivan Sergienko, Anatoliy Yakuba</i> SCIT — Ukrainian Supercomputer Project .....	63
<i>Valeriy Koval, Yuriy Kuk</i> New Knowledge Obtaining in Structural-predicate Models of Knowledge .....	69
<i>Valeriy Koval, Sergey Ryabchun, Volodymyr Savyak, Anatoliy Yakuba</i> Cluster Management Processes Organization and Handling .....	78
<i>Alexander Kleshchev, Irene Artemjeva</i> An Analysis of Some Relations Among Domain Ontologies .....	85
<i>Alexander Kuzemin, Darya Fastova, Igor Yanchevsky</i> Methods of Adaptive Extraction and Analysis of Knowledge for Knowledge-base Construction and Fast Decision Making .....	93
Table of Contents of Volume 12, Number 1 .....	100

### Table of Contents of IJ ITA Volume 12, Number 2

<i>Cristina Hernández, Juan Castellanos, Rafael Gonzalo, Valentín Palencia</i> Neural Control of Chaos and Applications .....	103
<i>Jesús Cardeñosa, Carolina Gallardo, Eugenio Santos</i> Bridging the Gap Between Human Language and Computer-Oriented Representations .....	111
<i>Víctor Giménez-Martínez, Carmen Torres, José Joaquín Erviti Anaut, Mercedes Perez-Castellanos</i> A New Approach for Eliminating the Spurious States in Recurrent Neural Networks .....	119



<i>Carmen Torres, Elena Castiñeira, Susana Cubillo, Victoria Zarzosa</i> A Geometrical Interpretation to Define Contradiction Degrees between Two Fuzzy Sets.....	127
<i>Karola Witschurke</i> A Workbench for Document Processing.....	135
<i>Elena Bolshakova, Igor Bolshakov, Alexey Kotlyarov</i> Experiments in Detection and Correction of Russian Malapropisms by Means of the WEB .....	141
<i>Alexander Kleshchev, Irene Artemjeva</i> A Mathematical Apparatus for Domain Ontology Simulation. An Extendable Language of Applied Logic .....	149
<i>Velina Slavova, Alona Soschen, Luke Immes</i> Information Processing in a Cognitive Model of NLP .....	157
<i>May Kornijchuk, Inna Sovtus, Eugeny Tsaregradskyy</i> Mathematical Model of Re-Structuring Complex Technical and Economic Structures .....	166
<i>Xenia Naidenova</i> DIAGaRa: An Incremental Algorithm for Inferring Implicative Rules from Examples.....	171
<i>Sergey Mostovoi, Vasiliy Mostovoi</i> Active Monitoring and Decision Making Problem .....	186
<i>Krassimir Markov</i> Building Data Warehouses Using Numbered Information Spaces .....	193
Table of Contents of Volume 12, Number 2 .....	200

## Table of Contents of IJ ITA Volume 12, Number 3

<i>Galina Rybina, Victor Rybin</i> Static and Dynamic Integrated Expert Systems: State of the Art, Problems and Trends .....	203
<i>Nadezhda Kiselyova</i> Application of Artificial Intelligence Methods to Computer Design of Inorganic Compounds.....	212
<i>Nadezhda Kiselyova, Victor Dudarev, Ilya Prokoshev, Valentin Khorbenko, Andrey Stolyarenko, Dmitriy Murat, Victor Zemskov</i> The Distributed System of Databases on Properties of Inorganic Substances and Materials .....	219
<i>Elena Revunova, Dmitri Rachkovskij</i> Training a Linear Neural Network with a Stable LSP Solution for Jamming Cancellation .....	224
<i>Olga Nevzorova, Julia Zin'kina, Nicolaj Pjatkin</i> Applied Problems of Functional Homonymy Resolution for Russian Language.....	231
<i>Anatoli Nachev</i> An Approach to Collaborative Filtering by ARTMAP Neural Networks.....	236
<i>Mikhail Bondarenko, Grigorij Chetverikov, Alexandr Karpukhin, Svetlana Roshka, Zhanna Deyneko</i> Synthesis Methods of Multiple-valued Structures of Language Systems .....	243
<i>Oleksii Mostovyi</i> Signal Processing under Active Monitoring .....	249
<i>Milen Nikolov, Vera Behar</i> Analysis and Optimization of Synthetic Aperture Ultrasound Imaging Using the Effective Aperture Approach....	257
<i>Alexander Kleshchev, Irene Artemjeva</i> A Mathematical Apparatus for Ontology Simulation. Specialized Extensions of the Extendable Language of Applied Logic.....	265
<i>Andriy Shelestov, Oleksiy Kravchenko, Michael Korbakov</i> Services for Satellite Data Processing .....	272

<i>Dimiter Dobrev</i>	
Formal Definition of Artificial Intelligence.....	277
<i>Elena Bolshakova</i>	
Programming Paradigms in Computer Science Education.....	285
<i>Evgeny Eremin</i>	
Educational Model of Computer as a Base for Informatics Learning.....	291
<i>Adil Timofeev</i>	
Adaptive Routing and Multi-Agent Control for Information Flows in IP-Networks.....	295
Table of Contents of Volume 12, Number 3 .....	300

Table of Contents of IJ ITA Volume 12, Number 4

<i>Vitaly Velichko, Victor Gladun, Gleb Gladun, Anastasiya Godunova, Yurii Ivaskiv, Elina Postol, Grigorii Jakemenko</i>	
Logical Models of Composite Dynamic Objects Control.....	303
<i>Boris Fedunov</i>	
The On-board Operative Advisory Expert Systems for Anthropocentric Object .....	308
<i>Lubomir Stanchev</i>	
Defining Network Activity Patterns Using First Order Temporal Logics.....	317
<i>Vassil Dimitrov, Khan Wahid</i>	
On the Error-Free Computation of Fast Cosine Transform .....	321
<i>Leonid Huliantskyi, Andrii Baklan</i>	
Optimization of ATM Telecommunication Networks .....	328
<i>Galina Bogdanova, Tsvetanka Georgieva</i>	
Analyzing the Data in OLAP Data Cubes .....	335
<i>Alexander Kleshchev, Irene Artemjeva</i>	
A Mathematical Apparatus for Domain Ontology Simulation. Logical Relationship Systems .....	343
<i>Andrey Shelestov, Alexander Lobunets, Michael Korbakov</i>	
Grid-enabling Satellite Image Archive Prototype for UA Space Grid Testbed.....	351
<i>Volodymyr Donchenko, Denys Serbaev</i>	
Representation of Neural Networks by Dynamical Systems .....	358
<i>Sergey Krivoi, Lyudmila Matvyeyeva, Wioletta Grzywacz</i>	
Automata-based Method for Solving Systems of Linear Constraints in $\{0,1\}$ .....	362
<i>David Burns, Richard Fallon, Phil Lewis, Vladimir Lovitskii, Stuart Owen</i>	
Verbal Dialogue versus Written Dialogue .....	369
<i>Yuri Sotskov, Nadezhda Sotskova, Leonid Rudoj</i>	
Learning Technology in Scheduling Based on the Mixed Graphs .....	378
<i>Agris Nikitenko</i>	
Robot Control Using Inductive, Deductive and Case Based Reasoning .....	386
Tables of Contents of IJ ITA – Volume 12 .....	396
Authors' Index .....	399
Table of Contents of Volume 12, Number 4 .....	400

## AUTHORS' INDEX

Irene	Artemjeva	85, 149, 265, 343	Alexander	Lobunets	351
Andrii	Baklan	328	Vladimir	Lovitskii	369
Vera	Behar	257	Krassimir	Markov	193
Yuriy	Belov	49, 57	Lyudmila	Matvyeyeva	362
Galina	Bogdanova	335	Martin	Mintchev	18
Igor	Bolshakov	141	Sergey	Mostovoi	186
Elena	Bolshakova	141, 285	Vasilii	Mostovoi	186
Mikhail	Bondarenko	243	Oleksii	Mostovyi	249
David	Burns	369	Dmitriy	Murat	219
Jesus	Cardenosa	111	Anatoli	Nachev	236
Juan	Castellanos	103	Xenia	Naidenova	171
Elena	Castineira	127	Olga	Nevzorova	231
Grigorij	Chetverikov	243	Charles	Newton Price	18
Susana	Cubillo	127	Eugene	Nickolov	35
Luis Fernando	de Mingo Lopez	5	Agris	Nikitenko	386
Renato	de Sobral Cintra	18	Milen	Nikolov	257
Zhanna	Deyneko	243	Stuart	Owen	369
Vassil	Dimitrov	321	Valentin	Palencia	103
Dimiter	Dobrev	277	Mercedes	Perez-Castellanos	119
Milena	Dobрева	12	Nicolaj	Pjatkin	231
Volodymyr	Donchenko	358	Dimitrina	Polimirova-Nickolova	35
Victor	Dudarev	219	Elina	Postol	303
Evgeny	Eremin	291	Ilya	Prokoshev	219
Jose Joaquin	Erviti Anaut	119	Zinoviy	Rabynovych	41
Richard	Fallon	369	Dmitri	Rachkovskij	224
Darya	Fastova	93	Elena	Revunova	224
Boris	Fedunov	308	Svetlana	Roshka	243
Alexander	Fish	29	Leonid	Rudoj	378
Carolina	Gallardo	111	Sergey	Ryabchun	63, 78
Tsvetanka	Georgieva	335	Victor	Rybin	203
Víctor	Gimenez-Martinez	119	Galina	Rybina	203
Francisco	Gisbert	5	Eugenio	Santos Menendez	5, 111
Gleb	Gladun	303	Volodymyr	Savyak	63, 78
Victor	Gladun	303	Denys	Serbaev	358
Anastasiya	Godunova	303	Ivan	Sergienko	63
Rafael	Gonzalo	103	Andrey	Shelestov	272, 351
Wioletta	Grzywacz	362	Velina	Slavova	157
Cristina	Hernandez	103	Alona	Soschen	157
Leonid	Hulianytskyi	328	Yuri	Sotskov	378
Roman	Iamborak	49, 57	Nadezhda	Sotskova	378
Nikola	Ikonomov	12	Inna	Sovtus	166
Luke	Immes	157	Lubomir	Stanchev	317
Yurii	Ivaskiv	303	Andrey	Stolyarenko	219
Grigorii	Jakemenko	303	Adil	Timofeev	295
Alexandr	Karpukhin	243	Sergiy	Tkachuk	49, 57
Valentin	Khorbenko	219	Carmen	Torres	119, 127
Nadezhda	Kiselyova	212, 219	Eugeny	Tsaregradskyy	166
Alexander	Kleshchev	85, 149, 265, 343	Vitaly	Velichko	303
Michael	Korbakov	272, 351	Khan	Wahid	321
May	Kornijchuk	166	David	Westwick	18
Alexey	Kotlyarov	141	Karola	Witschurke	135
Valeriy	Koval	63, 69, 78	Orly	Yadid-Pecht	29
Oleksiy	Kravchenko	272	Anatolij	Yakuba	63, 78
Sergey	Krivoi	362	Igor	Yanchevsky	93
Yuriy	Kuk	69	Victoria	Zarzosa	127
Alexander	Kuzemin	93	Victor	Zemskov	219
Phil	Lewis	369	Julia	Zin'kina	231

---



---

**TABLE OF CONTENTS OF VOLUME 12, NUMBER 4**

Logical Models of Composite Dynamic Objects Control.....	303
<i>Vitaly Velichko, Victor Gladun, Gleb Gladun, Anastasiya Godunova, Yuri Ivaskiv, Elina Postol, Grigori Jakemenko</i>	
The On-board Operative Advisory Expert Systems for Anthropocentric Object.....	308
<i>Boris Fedunov</i>	
Defining Network Activity Patterns Using First Order Temporal Logics.....	317
<i>Lubomir Stanchev</i>	
On the Error-Free Computation of Fast Cosine Transform.....	321
<i>Vassil Dimitrov, Khan Wahid</i>	
Optimization of ATM Telecommunication Networks.....	328
<i>Leonid Hulianytskyi, Andrii Baklan</i>	
Analyzing the Data in OLAP Data Cube.....	335
<i>Galina Bogdanova, Tsvetanka Georgieva</i>	
A Mathematical Apparatus for Domain Ontology Simulation. Logical Relationship Systems.....	343
<i>Alexander Kleshchev, Irene Artemjeva</i>	
Grid-enabling Satellite Image Archive Prototype for UA Space Grid Testbed.....	351
<i>Andrey Shelestov, Alexander Lobunets, Michael Korbakov</i>	
Representation of Neural Networks by Dynamical Systems.....	358
<i>Volodymyr Donchenko, Denys Serbaev</i>	
Automata-based Method for Solving Systems of Linear Constraints in $\{0,1\}$ .....	362
<i>Sergey Krivoi, Lyudmila Matvyeyeva, Wioletta Grzywacz</i>	
Verbal Dialogue versus Written Dialogue.....	369
<i>David Burns, Richard Fallon, Phil Lewis, Vladimir Lovitskii, Stuart Owen</i>	
Learning Technology in Scheduling Based on the Mixed Graphs.....	378
<i>Yuri Sotskov, Nadezhda Sotskova, Leonid Rudoj</i>	
Robot Control Using Inductive, Deductive and Case Based Reasoning.....	386
<i>Agris Nikitenko</i>	
Tables of Contents of IJ ITA – Volume 12.....	396
Authors' Index.....	399
Table of Contents of Volume 12, Number 4.....	400