I T H E A

International Journal

INFORMATION TECHNOLOGIES & KNOWLEDGE

2009 Volume 3 Number 1

International Journal
INFORMATION TECHNOLOGIES & KNOWLEDGE
Volume 3 / 2009, Number 1

Editor in chief: **Krassimir Markov** (Bulgaria)

IJ ITK is official publisher of the scientific papers of the members of the ITHEA International Scientific Society

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

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

International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE" Vol.3, Number 1, 2009

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: **ITHEA®**
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com
Printed in Bulgaria

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

PREFACE

Intelligo ut credam !

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

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

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

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

This Volume 3/2009 of the IJ ITK contains XX invited papers.

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

The success of IJ ITK belongs to the whole of the ITHEA International Scientific Society. We express our thanks to all authors, editors and collaborators who had developed and supported the International Journal on Information Technologies and Knowledge.

XV^{-th} anniversary of the International Prize "ITHEA"!

The **International Prize "ITHEA"** for outstanding achievements in the area of information theories and applications has been established in 1995.

For fifteen years, 41 leading scientists of the ITHEA International Scientific Society have been awarded by the "ITHEA" Prize.

Congratulations to Prof. Anatoliy Danilovich Krissilov who has been awarded by the "ITHEA" Prize for 2009 year.

Awarded Scientists till 2009:

1995	*Sandansky*	*K. Bankov, P. Barnev, G. Gargov, V. Gladun, R. Kirkova, S. Lazarov, S. Pironkov, V. Tomov*
1996	*Sofia*	*T. Hinova, K. Ivanova, I. Mitov, D. Shishkov, N. Vashchenko*
1997	*Yalta*	*Z. Rabinovich, V. Sgurev, A. Timofeev, A. Voloshin*
1998	*Sofia*	*V. Jotsov*
1999	*Sofia*	*L. Zainutdinova*
2000	*Varna*	*I. Arefiev, A. Palagin*
2001	*St.Peterburg*	*N. Ivanova, V. Koval*
2002	*Primorsko*	*A. Milani, M. Mintchev*
2003	*Varna*	*T. Gavrilova, A. Eskenazi, V. Lozovskiy, P. Stanchev*
2004	*Varna*	*B. Kokinov, T. Vamos*
2005	*Varna*	*M. Dobreva, L.F. de Mingo, N.Zagoruiko, A.Zakreuskii*
2006	*Varna*	*J. Castellanos, G. Totkov*
2007	*Kiev*	*V. Donchenko*
2008	*Varna*	*Vladimir Lovitskii, Orly Yadid-Pecht*
2009	*Varna*	*A. Krissilov*

Krassimir Markov

IJ ITK Founder and Editor in chief

EXPERIMENTAL SUPPORT OF SYNTACTIC COMPUTATION BASED ON SEMANTIC MERGE OF CONCEPTS

Velina Slavova, Alona Soschen

Abstract: *Linguistic theory, cognitive, information, and mathematical modeling are all useful while we attempt to achieve a better understanding of the Language Faculty (LF). This cross-disciplinary approach will eventually lead to the identification of the key principles applicable in the systems of Natural Language Processing. The present work concentrates on the syntax-semantics interface. We start from recursive definitions and application of optimization principles, and gradually develop a formal model of syntactic operations. The result – a Fibonacci-like syntactic tree – is in fact an argument-based variant of the natural language syntax. This representation (argument-centered model, ACM) is derived by a recursive calculus that generates a mode which connects arguments and expresses relations between them. The reiterative operation assigns primary role to entities as the key components of syntactic structure. We provide experimental evidence in support of the argument-based model. We also show that mental computation of syntax is influenced by the inter-conceptual relations between the images of entities in a semantic space. This work represents a further step in the formal description of the observed syntax-semantics dependencies. The assumption is made that the syntax-semantic interface is best explained as a particular operation, Merge that applies at both semantic and syntactic levels. The resulting formal description of the stages of syntactic treatment complies with the results of the experiment.*

A formal model of syntactic operations is developed starting from recursive definitions and application of optimization principles. The result – a Fibonacci-like syntactic tree – is in fact an argument-based variant of the natural language syntax.

Keywords*: natural language, mathematical modeling, cognitive modeling*

ACM Classification Keywords*: I.2 Artificial Intelligence, 1.2.0. Cognitive simulation*

Introduction

We use mathematical formalism of Generalized Nets to develop a stage-simulating model of NLP. This formal approach allows a more exact representation of information flows during the stages of processing, expressed as the transitions Z_1–Z_{29} of the Net (Slavova 2004). The analyses performed on this basis suggest that information treatment consists of the operations that use two types of Long Term Memory knowledge (syntactic and semantic) in parallel. As an example, this is the case of transition Z_{27}, which expresses the stage when the system builds the syntactic structure of a sentence after its last word-form was stored in Working Memory (figure 1.). A detailed examination of the incoming information flow allows us to suggest that the procedure, running on Z_{27}, must use semantic and syntactic knowledge in parallel. We assumed that **syntactic structure is better clarified** when it receives semantic justification.

For further analyses, the two types of knowledge stored in Long Term Memory were modeled by means of a database structure that shows the interconnection of syntactic rules, semantic primitives, and semantic operators (Slavova, Soschen, Immes, 2005). The assumption was that language units (word-forms) have images as semantic primitives such as “concepts”, “attributes”, “events” etc, and that grammatical rules comply with semantic operations on these primitives. This formalization of the Language as a "joint" Information System was used to study a particular language rule - secondary predication in Russian[1]. This rule was modeled by means of the formal approach described above. That led to a coherent and well-defined formal procedure and confirmed that the rule entails operations on semantic primitives.

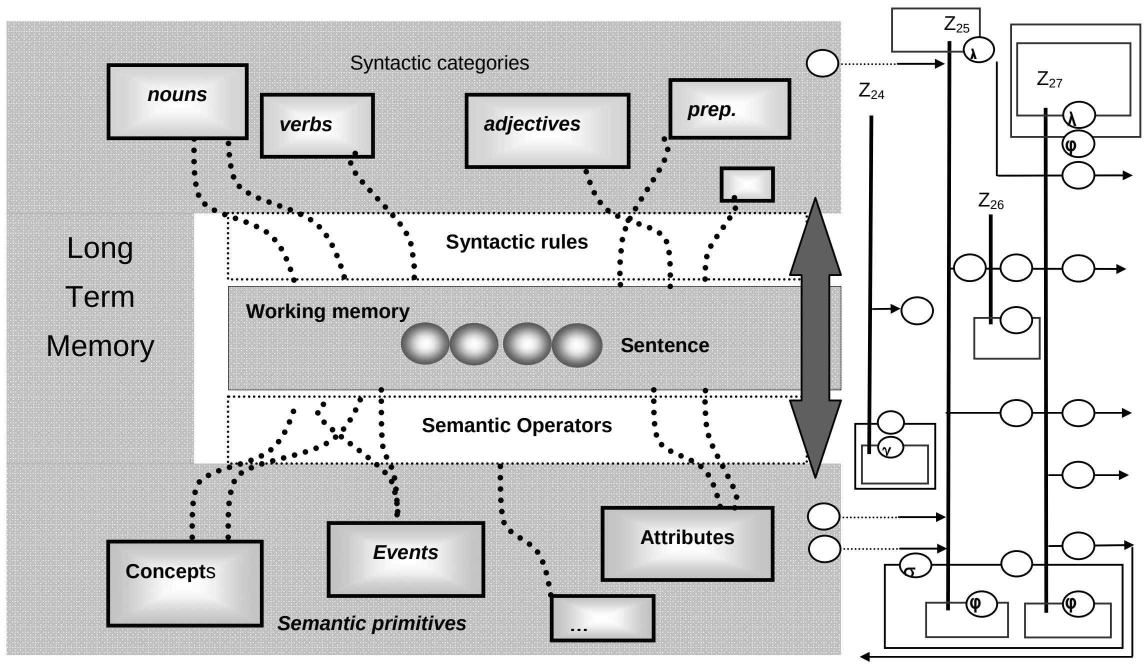


Figure 1. Information treatment of a sentence, based on language and semantics

Further efforts are put forward to obtain the proof that **semantic knowledge** and **syntax** are interrelated. The question so far is how syntax is related to operations on semantic primitives – concepts, events, attributes, etc. This is one of the most important questions in contemporary linguistics and cognitive science.

Syntax as Computation

Following one of the widely accepted linguistic theories, the key component of Faculty of Language (FL) is a computational system (narrow syntax) that generates internal representations and maps them into the conceptual-intentional interface by the (formal) semantic system (Hauser et al., 2002). There is a consensus that the core property of FL is *recursion*, which is attributed to narrow syntax. In other words, the process of mental generation of syntactic structures relies on the capacity of the human brain to perform specific operations in compliance with the principles of efficient computation. The claim in the recent theories is that this computation is based on a primitive operation that takes already constructed objects to create a new object. This basic operation, called "Merge", provides a "language of thought", an internal system to allow preexistent conceptual resources to construct expressions (Chomsky, 2006). Although these questions receive a lot of attention, there

[1] The linguistics theories don't provide a consistent explanation of Secondary Predication in Russian.

are no convincing proposals yet concerning the precise type of resources on which such computation is performed in a recursive manner to build syntactic structures.

Following from the above, the study of syntactic recursion by mathematical means may provide valuable insights into the principles underlying the human language. One step in this direction was provided in Slavova and Soschen (2007). Syntactic structures, presented in the traditional sense of Chomskyan theory (Bare Phrase Structures, XP-structures), were re-defined in terms of finite recursive binary trees. The “traditional syntactic tree” does not correspond to the finite nature of a sentence; consequently, it cannot be defined recursively as a finite object. Another reason to introduce this modification is to build a structure that complies with the principles of optimization, namely with the principle of efficient growth (Soschen 2006, 2008). The tree was modified; the nodes related to syntactic role of verbs were discarded. The structure obtained in this way is a tree of Fibonacci (figure 2. a).

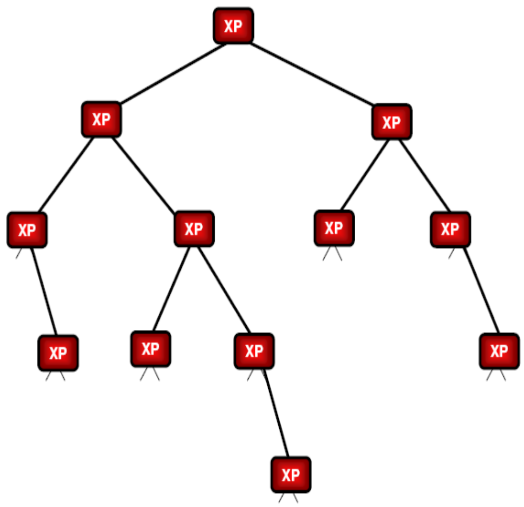


Figure. 2a.

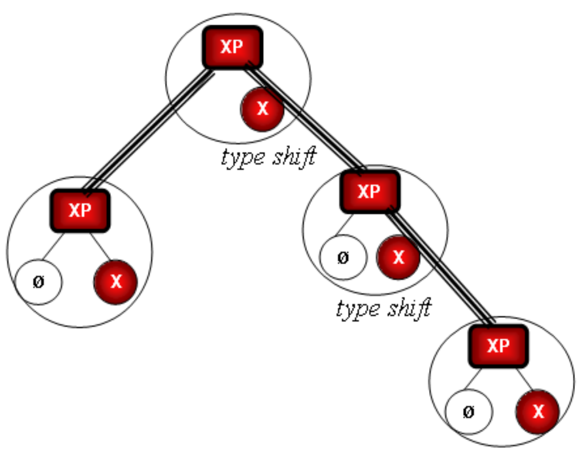


Figure. 2. b.

This tree can be seen as is an operator – it “performs” a bottom-up Merge (figure 2.b.), its nodes are the results of Merge. In the model under development, XPs are *sets*, Xs are ‘unbreakable’ *entities*, and Merge can be applied to two non-equivalent substances (the tree has ordered nodes). These formal transformations of the traditional tree result in a structure that incorporates two operations of fundamental importance in the syntactic model. The first is "Ø-Merge", operation that takes place at the point where Xs as initial substances form *singleton sets,* ready for further syntactic computation. The second is *type-shift,* which results in a transition from *sets* (XPs) to *entities* Xs and expresses a property of the dual mental representation of XP as either consisting of two separate elements or as an 'unbreakable' whole (part of a larger unit).

The Fibonacci-like tree shows the patterns of relating arguments (Soschen 2006, 2008). An important question is the height h of the XP Fibonacci-tree, since it refers directly to the memory, necessary for the computation. The tree is a recursive object; the same patterns of Merge are repeated at its levels. It is easy to show that merge-patterns start to reiterate when h>3 and that any tree with h>3 can perform more than one merge-pattern. We defined the tree with h=3 as the basic tree (fig. 2.b). We interpret its properties as follows: the basic tree defines the maximal number of Xs that can be merged in a procedurally unambiguous way. It could be suggested that this structure is determined in the same way as the number of nodes and relations that can be treated by the human brain within a semantically meaningful argument space. The tree represents a bare (label-free) syntactic structure that has no lexical input; what it has are the paths that connect smaller units in order to produce a larger meaningful unit. We called the tree in (fig. 2.b) "the Argument-Based Syntactic Tree".

According to the hypothesis put forward in Soschen (2005, 2006, 2008), a general rule governing efficient growth applies in syntax in such a way that minimal syntactic constituents incorporate arguments (*agent, recipient, theme*) which are related to each other. In the Fibonacci-tree model, the type of merge configuration determines the type of relation between arguments. The maximal configuration (fig. 3.d) corresponds to thematic roles *agent*, *recipient*, and *theme*. The "syntactic meaning" of the schemes in (fig. 3) corresponds to configurations offered in (Soschen 2006, 2008). These schemes represent all possible configurations and relations between arguments in the human theta-role Semantic Space. Carnie (2006) shows convincingly that the number of arguments in a thematic domain is necessarily limited to three, a fact that has not found an explanation in linguistics so far. The model under development suggests that the number of arguments is limited in a particular way in compliance with the principles of efficient growth, which are, in our terms, the principles of efficient computation as well.

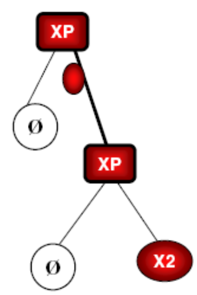


Figure 3.a. Infinite iteration: Mary, Mary…

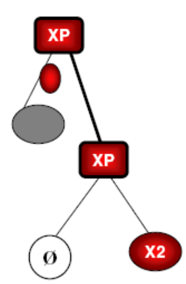


Figure 3.b. Mary in *Mary smiles*.

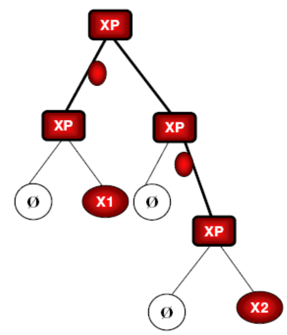


Figure 3.c. Two arguments Mary, John in Mary loves John.

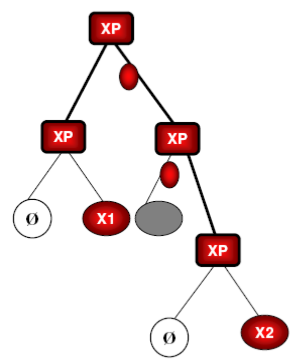


Figure 3.d. Three arguments Mary, John, apple in Mary gave John an apple.

Of importance to linguistic theory is our proposal that the argument-based model of syntax has a fundamental character. This model shows that syntax utilizes recursive calculus to connect *arguments* and express *relations* between them. The argument-based model assigns a primary syntactic role to *entities*, usually expressed as nouns. This viewpoint is in contrast with *verb*-centered models of syntax. Our efforts are focused on the experimental evidence that supports the argument-based model. The difficulty of designing an appropriate experiment is that mental computation runs on a deep (pre-linguistic) level and cannot be captured on the lexical level by a standard experiment. One possible way to extract some information about the primary mechanisms is to force the mental system to solve ambiguities on the lexical level and to analyze the system's response.

Experimental Design

Bulgarian is the only Slavic language which, during the last 10 centuries, has undergone a transition from synthetic to analytical language. Prepositions replaced case-flections, and a suffixed definite article appeared. One interesting result of the transition is that the Genitive and Dative cases are both expressed by means of the preposition *'на'* (na). "Na" has several meanings: to, of, on. Our experiment is based on the following two meanings of "Na":

1. *Of – meaning* (whose, Slavonic Genitive)

The X | *на* | *the* Y   means "the X of the Y" i.e. "the Y's X", as in:

The X	На	The Y	
Къщата	На	Кучето	
The house	Of	The dog	The dog's house

2. To – meaning (to whom, Slavonic Dative)

Subject Verb *на* | *the* Y   means that the subject S acts To the Y. For transitive verbs, на assigns the syntactic role of a *Recipient*:

S Verb O		The Y (Recipient)
Той донесе стол	На	Директора
He brought a chair	To	the director

In the example above, Object is not marked with an article. Such sentences always have the meaning S-(V)-O-R (three arguments: agent, theme, and recipient).

When the Object is marked with an article, the sentence becomes:

Subject | Verb | *the X (*Object) | *на* | *the Y* .

and its second part fits the Genitive construction *the* X | *на* | *the* Y . In result, the available grammatical rules of the language assign to the noun Y two possible roles:

1. Subject | Verb | *the* X (Object) | *to* | *the Y (*Recipient). S-(V)-O-R, Resipient (1)

2. Subject | Verb | *the* X (Object) | *of* | *the* Y (Possessor). S-(V)-O-of-P, Possessor (2)

In such sentences, preposition *на* indicates that the noun that follows it is either Recipient (argument), or it is the object's owner/ Possessor. The difference between these two interpretations is crucial, as the basic syntactic structure of two sentences is completely different - in the former, there are three arguments, and in the latter, there are two (corresponding respectively to the trees on fig. 3.d and 3.c). In Bulgarian, all the sentences of type:

Subject | Verb | *the* Object | *на* | *the* Y.

are ambiguous: they assign two different meanings to Y - *Recipient* and *Possessor*.

In normal listening or reading-comprehension conditions, native Bulgarian speakers interpret one of these meanings depending on the context. The sentence "Mary gave the book на the boy." in the context "Mary entered holding a book and she saw a boy" is interpreted as "Mary gave the book to the boy." And, in the context "The boy left his book. Mary was asked about the book." the very same sentence is interpreted as "Mary gave the boy's book to someone else." Speakers of Bulgarian are never mistaken about the conveyed meaning. However, as our experiment has shown, they are not even aware of the existence of the two meanings. It appears that in the cognitive space such "*на*-sentence" acts as a Necker Cube – one may "see it" in either of the two ways. The context makes one of the meanings explicit, while the subjects are not aware of the other meaning. And, in fact, as is the case with Necker's Cube, if one concentrates long enough on an isolated *на*–sentence, one will discern that it has two meanings.

Our goal is to study the mechanisms of mental computation of the syntactic structure of an isolated sentence, with regard to the role of the verb and the arguments.

1. If the assumption is correct that the argument-centered computation is the key to mental operations, an isolated *на*-sentence will be constructed by assigning to Y the role of Recipient.
2. The *на*-sentences are ambiguous; if the role of *entities* (nouns in this case) is primary, semantic relations between their images in the conceptual nets will influence the final result of the syntactic computation.

Experiment

In what ways an isolated *на*–sentence is interpreted? We prepared 13 examples of *на*-sentences (Table 1). Each of these sentences has an argument that conveys either of the two meanings – Recipient (Rc) vs. Possessor (Ps). All the verbs used in the test examples are transitive and allow Recipient. All the sentences can exist as complete sentences without Possessor and without Recipient. The verbs are in the past tense, Perfective form.

Table 1.

200.Ex	*Иван* *Ivan*	*Продаде* *Sold*	*Къщата* *The house*	*На* *to/of*	*баща си* *his father*
201.Ex	*Мария* *Mary*	*Продаде* *Sold*	*Колата* *the car*	*На* *to/of*	*Съседката* *the neighbour*
202.Ex	*Михаил* *Mihail*	*Продаде* *Sold*	*Къщата* *The house*	*На* *to/of*	*съседа си* *his neighbour*
203.Ex	*Елена* *Elena*	*Продаде* *Sold*	*Къщата* *The house*	*На* *to/of*	*Кучето* *the dog*
204.Ex	*Анна* *Anna*	*Продаде* *Sold*	*Ябълките* *The apples*	*На* *to/of*	*Момчето* *the boy*
211.Ex	*Анна* *Anna*	*Подаде* *Gave*	*Стола* *the chair*	*На* *to/of*	*Директора* *the director*
212.Ex	*Петър* *Peter*	*Донесе* *Brought*	*Стола* *the chair*	*На* *to/of*	*Директора* *the director*
220.Ex	*Мария* *Mary*	*Показа* *Showed*	*Колата* *the car*	*На* *to/of*	*Съседката* *the neighbour*
221.Ex	*Иван* *Ivan*	*Показа* *Showed*	*Пътеката* *the wolk*	*На* *to/of*	*Баща си* *his father*
222.Ex	*Петър* *Peter*	*Показа* *Showed*	*Къщата* *The house*	*На* *to/of*	*Баща си* *his father*
231.Ex	*Кумчо Вълчо* *The Big Bad Wolf*	*Продаде* *Sold*	*Къщата* *The house*	*На* *to/of*	*Кучето* *the dog*
232.Ex	*монтьорът* *The fitter*	*Показа* *Showed*	*Колата* *The car*	*На* *to/of*	*Съседката* *the neighbour*

We need to find out which of the two meanings of these isolated sentences is obtained FIRST, i.e. in the most natural way. That can provide information about the mechanisms of mental computation of the basic syntax.

The difficulty in designing an efficient experiment is that when asked to explain the meaning of such a sentence, subjects usually reply by repeating the very same sentence. For them, in the first moment, the sentence has only one meaning that can be put into words in one particular way only. The subjects do exactly what they were asked to do: they express the meaning by using words. Further efforts to make them reveal the meaning make them focus on the sentence for a longer period of time. As a result, they discover that the sentence has one more meaning, and they report that the sentence can mean two different things.

This difficulty was overcome in a tricky way. We used the fact that sentence structure, including word order, is exactly the same in French. The crucial difference is that the preposition *на* is translated in French as “à” (to) for the Recipient-meaning and as “de” (of) for the Possessor-meaning.

The subjects of our experiment were the students in the masters program of the Francophone Institute for Management in Sofia [1], all of them fluent speakers of French. The subjects, 62 students with different backgrounds (economists, sociologists, biologists, linguists, engineers etc.), were: native speakers of Bulgarian - 39, of Ukrainian - 6, of Rumanian – 5, of Russian - 3, of Georgian – 3, of Albanian – 3, of Macedonian – 2, and of Arabic – 1. Some of the non-native Bulgarians spoke Bulgarian fluently, some were less fluent.

The statements in Bulgarian were presented in a written form to the subjects, on small separate pieces of paper, with the only instruction “Translate into French”. It was done at the end of regular classes, under circumstances implying that “it is not something you should worry about, do it speedily”.

Each statement was presented to 10-12 different subjects. Each subject was given 2 different statements in a random manner, while the statements did not contain the same verb or the same noun. The 23 non-native Bulgarian speakers could ask the experimenter about the meaning of Bulgarian words. There were a few questions about the meaning of “монтьор” (fitter), “тапицер” (upholster) and “пътека” (path) as well as about the corresponding French-tense of the verbs (Past-perfect forms are translated with “passé composé”). There were no questions about the meaning of *на*.

The 124 written translations of the test statements were stored in a database. Table 2 contains the proportion of the Recipient- and Possessor-meanings assigned to each statement (Of% and To%).

This experimental design was successful in the sense that only 4 subjects, native Bulgarian speakers, became aware that a given sentence has 2 meanings. It is interesting that some of these subjects noticed the double meaning of one of the statements that they had to translate, but not of the other. They were asked to put down the two possible translations in the order in which the meanings came to their minds, and only the first one was taken into account for further analyses.

The results in Table 2 show that, in spite of the “Necker’s cube property” of each statement, one of its possible meanings is interpreted by the subjects more often than the other. The second observation is that for some statements the preferred interpretation is the Recipient-meaning and for others – the Possessor-meaning**.** The third observation is that these changes do not depend on the verb. For one and the same verb, the interpretation “switches” from one to the other meaning. For example, as one can see in Table 2, “Sold” appears in statements varying from 100% of Recipient-meaning, to 100 % of Possessor-meaning.

---

[1] Institut de la Francophonie pour l’Administration et la Gestion - Sofia

Table 2.

	Subject	Verb	Object	Ha	Y	Of%	to%	Tendency
204.Ex	Anna	Sold	The apples	To/of	the boy		**100**	Y = Recipient
202.Ex	Mihail	Sold	The house	To/of	his neighbor	29	71	Y -> Recipient
201.Ex	Mary	Sold	The car	To/of	the neighbor	30	70	Y -> Recipient
231.Ex	The Big Bad Wolf	Sold	The house	To/of	the dog	33	67	Y -> Recipient
200.Ex	Ivan	Sold	The house	To/of	his father	67	33	Y -> Possessor
203.Ex	Elena	Sold	The house	To/of	the dog	**100**		Y = Possessor
221.Ex	Ivan	showed	The path	To/of	his father		**100**	Y = Recipient
220.Ex	Mary	showed	The car	To/of	the neighbor	11	89	Y = Recipient
222.Ex	Peter	showed	The house	To/of	his father	33	67	Y -> Recipient
232.Ex	The fitter	showed	The car	To/of	the neighbor	50	50	Equivalence
211.Ex	Anna	gave	The chair	To/of	the director		**100**	Y = Recipient
212.Ex	Peter	brought	The chair	To/of	the director	13	88	Y = Recipient
233.Ex	The upholster	brought	The chair	To/of	the director	50	50	Equivalence

Based on the available experimental data (at least ten trials for each statement from different subjects), we assume that the experiment has captured some major tendencies in the interpretation of the test statements. This experiment allows us to further explore the principles of mental operations underlying interpretation of the basic syntactic argument structure. So far, a linguistic theory that would explain the observed tendencies in obtaining some particular result, “computed” by the subjects, has not been developed. Our experiment has shown that the explanation can be provided by using the argument-oriented model derived in compliance with the principles of efficient computation.

Analyses of Experimental Results

The experimental results show that the interpretation of the syntactic structure depends on entities (in this case, nouns). The verb itself does not predetermine the type of structure: either S-(V)-O-R (three arguments) or S-(V)-OofY (two arguments). Many of the contemporary linguistic theories mostly consider predicate-based and verb-centered syntactic structures. Actually, if the verb does not allow a recipient, the syntactic structure of the *на*-sentence is calculated as S-(V)-O of Y.

Suppose that mental calculus depends solely on the type of the verb. Then in the cases where the verb allows Rc, на would ALWAYS imply a S-(V)-O-R structure. But that is clearly not the case in the last four examples, given in Table 3 (where the examples are arranged by the “captured from subjects meaning”) :

As it is shown in Table 3, when the verb allows a Recipient, *на* implies preferably, but not necessarily the structure S-(V)-O-R (three arguments). The noun Y selects the Rc role in most cases. If mental operations were not dependent on the calculus which relies on the arguments as primary substances, all the statements of the experiment would be with around 50% interpretation of Y as Rc and 50% - Y as Ps.

We may suppose that the argument-centered representation of syntax is the key to syntactic analyses.

Table 3

Subject	Verb	Object	Ha	Y	of%	To%	Tendency
Anna	Sold	the apples	to/of	the boy		100	Y = Recipient
Ivan	showed	the path	to/of	His father		100	Y = Recipient
Anna	Gave	the chair	to/of	The director		100	Y = Recipient
Mary	showed	the car	to/of	the neighbor	11	89	Y = Recipient
Peter	brought	the chair	to/of	The director	13	88	Y = Recipient
Mihail	Sold	the house	to/of	his neighbor	29	71	Y -> Recipient
Mary	Sold	the car	to/of	the neighbor	30	70	Y -> Recipient
The Big Bad Wolf	Sold	the house	to/of	The dog	33	67	Y -> Recipient
Peter	showed	the house	to/of	His father	33	67	Y -> Recipient
the fitter	showed	the car	to/of	the neighbor	50	50	Equivalence
the upholster	brought	the chair	to/of	The director	50	50	Equivalence
Ivan	Sold	the house	to/of	His father	67	33	Y -> Possessor
Elena	Sold	the house	to/of	The dog	100		Y = Possessor

The next question is: if the argument S-(V)-O-R structure is calculated first, what are the reasons that lead the calculus to take another route and assign a S-(V)-O of Y structure to a similar sentence? Our assumption is that the sentence is kept in working memory (figure 1.) and that the final “solution” about basic syntactic roles is assigned to all its parts after semantic verification. If that was not true, the word order would be the key factor in the syntactic computation and the observed differences in the interpretation would not appear.

Let us analyze why the statement:

Elena Sold The house *to/of* The dog. | 100% of | Y = Possessor,

is interpreted as having S-(V)-O of Y structure. The reason for that seems very clear: the noun dog is rejected as Rc of “sold”. The noun takes upon itself the role of the owner of the house. If this is the right mechanism, it is sufficient to provide “the dog” with the possibility to be the Rc of the house, or to modify a noun: “Elena sold the house to a dog-buyer”.

The argument-centered syntactic model attests to the fact that syntactic relations depend on the relations between concepts that exist in the semantic space. In fact, as the experimental results show, it is sufficient to replace the subject noun with the one that can be related to the dog as a buyer in a fairy tale context:

The Big Bad Wolf Sold The house *to/of* The dog | To 67% | Y -> Recipient

This result indicates that mental calculus takes into consideration not only the meaning of the noun but also the relations between the nouns. Thus:

Ivan	Showed	The path	*To/of*	His father	To	100%	Y = Recipient
Ivan	Sold	The house	*To/of*	His father	67%	Of	Y -> Possessor

The three possible arguments of both sentences correspond to concepts that exclude relations such as "fathers have paths" or "sons sell houses to their fathers". Note that sentences reveal the relations between all the three of the arguments. The predominant meaning in the semantic space of the second sentence is 'fathers have houses and sons operate their father's property'.

These dependencies between the basic concepts expressed as Subject and Object are shown as two pairs of statements below:

Subject	Verb	Object	Ha	Y	of%	To%	Tendency
Mary	Showed	The car	to/of	The neighbor	11	89	Y = Recipient
The fitter	Showed	The car	to/of	The neighbor	50	50	Equivalence
Peter	Brought	the chair	to/of	The director	13	88	Y = Recipient
The upholster	Brought	the chair	to/of	The director	50	50	Equivalence

When Mary shows the car, she shows it TO the neighbor; when the fitter shows the car, there is a high probability that this is the neighbor's car. In the semantic space, fitters operate on cars, while neighbors have cars. That same tendency is observed in the second in pair (upholsters and a director's chair). Once again, argument structure is influenced by the inter-conceptual relations.

These examples provide evidence about the nature of the primary elements - participants in mental operations. It becomes clear that syntactic computation depends on the meaning of the nouns and inter-conceptual relations.

AGN Tracking

AGN is a Generalized Net model, developed for simulating the cognitive process of natural language comprehension (Slavova 2004). This formal approach allows a more exact representation of information flows during the stages of processing, expressed as the transitions Z_1—Z_{29} of the Net. AGN has been used as formalism in several studies in cognitive linguistics. The detailed examination (Slavova, Soschen (2005)) of the information flow has led to the suggestion that the procedures, running on Z_{25} and Z_{27}, must use semantic and syntactic knowledge in parallel (figure 4). Contemporary linguistic theories don't provide the necessary theoretical bases for the formalization of the process (the most part of them don't agree that syntax is related to semantics). The presented work aims to give further development of AGN by analyzing the procedures which perform simultaneously in the language and the semantic space. Transition Z_{25} (figure 4) simulates two processes, which run in parallel - the activation of the semantic space σ (the Semantic net NSet stays on l_{70}) by the message word-forms Wi and the detection of grammatically related word-chains in the sentence.

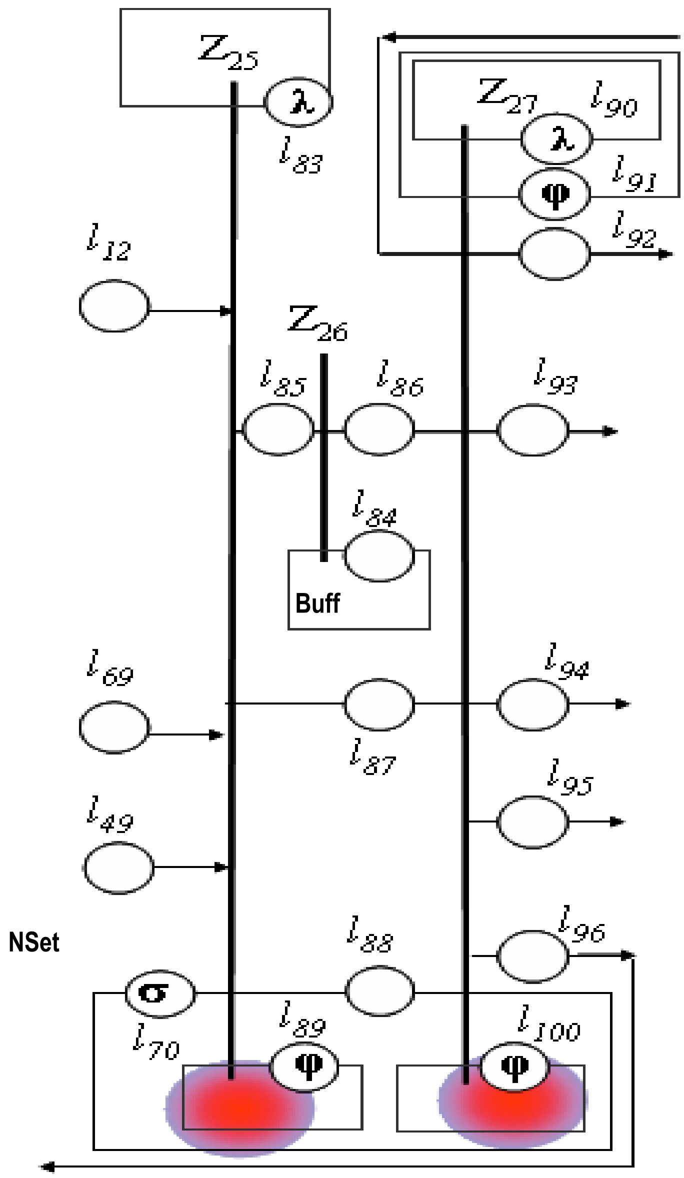


Figure 4. Transitions, based on language and semantics

It is supposed that the cognitive system first assembles a fractional representation of the sentence-meaning structure (coupled words for example).

The incoming information flow for Z_{25} is:

l_{12} – Wi+GrFtrs - word-forms Wi, with their grammatical features GrFtrs;

l_{49} - Ntct - the list of nodes of the semantic NSet, corresponding to Wi and the couples WNk: (Wi, NSetj), representing the word-forms Wi, assembled with their corresponding nodes of NSet.

l_{69}, - NtSBlist - the list of attributes of the concepts

The running on Z_{25} procedures are:

TreeBranches on I_{83} (λ-token), which obtains a partial syntactic tree:

“ParSynStr = TreeBranches (NSet, GrFtrs) (3)

SemA (φ-token) on I89 which stores activation ANet in the semantic net NSet:

ANet = SemA (Ntct, NtSBlist) (4)

Transition Z_{26} represents a Working Memory buffer (Buff). It stores the W_i + GrFtrs, with their corresponding semantic nodes – the couples WN_k: (W_i, $NSet_j$) and represents the “lexical memory”, storing the words with their meanings.

Transition Z_{27} expresses the mental process of analyzing the entire sentence after its last word-form has been perceived. It employs from I_{88} the activation ANet of the semantic net NSet as well as its structure. The following procedures are running on Z_{27}:

λ-token - Procedure “*Parse*” on place I_{90} with *“Syntax structure discovery”*

φ-token - Procedure “*Comp*” “Comparing semantics and syntax” on I_{91}

The previous analyses (Slavova, Soschen 2005) led to the supposition that the procedure *Parse* entails a “***Merge concepts***” operator, performed in the semantic space on Z_{27}. This supposition is developed in the analysis here. The information processing on transitions Z_{25} and Z_{27} is analyzed for the statements of type:

| X | | Verb | | the Y | | *на* | | the Z |.

Z_{25}: Transition Z_{25} receives from its “language” input I_{12} the word-forms W_i with their grammatical features GrFtrs, such as gender, plural, articles etc. (Table 4).

Following grammatical features, procedure TreeBranches assembles “X” with the verb “Verb” as well as the preposition Na with the noun Z which follows it.

TreeBranches (0, GrFtrs) = *ParSynStr* ={{X, Verb}, {*на*, Z}}} (5)

Table 4.

Wi:	**GrFtrs:**		
X	Noun	3 pers.	Sing.
Verb	Verb	3 pers.	Sing.
Y	Noun		/article
Na	Preposition		
Z	Noun		/article

Table 5.

NSeti semantic images:
X* - the concept of X
V*- the image of the achievement (?O, R)
Y* - the concept of Y
To* Of*
Z* - the concept of Z

The words W_i have corresponding images in the semantic net NSet, given in Table 5. These nodes and their related nodes are activated by the procedure *SemA*.:

$$ANet =\{ X^*, V^*, Y^*, (To^*, Of^*), Z^*\} \cap \{\{rel\ X^*\}, \{rel\ V^*\}, \{rel.\ Y^*\}, \{rel\ Z^*\}\}\} \quad (6)$$

Z_{27}: To the input of Z_{27} come results from:

I_{86} – the content of Buff: W_i + GrFtrs (Table 2) and WN_k: the couples WN_k: (W_i, $NSet_j$)

I_{87} – the edges of the partial tree "ParSynStr = {{X, Verb}, {*на*, Z}}"

I_{88} – The semantic net NSet with the activated nodes ANet.

Transition Z_{27} has to perform the language procedure Parse(Buff, ParSynStr) and to perform the semantic check Comp (Parse(), ANet).

Syntactic Procedure with Semantic Merge

Procedure Parse on I_{90} uses knowledge of syntactic rules. The assumption is that it runs by executing operation "Semantic Merge", based on the information, available on the inputs of Z_{27}.

Semantic Merge (M) is further modeled as a binary operation, performed in sequential progression between the concepts X*, Y*and Z* inside of a sentence with Verb V*. The main assumption here is that Semantic Merge is in compliance with the principles of the argument-based syntactic model. The result of Semantic Merge consists in temporal semantic images μ, which stay in working memory till the end of the syntactic treatment.

The syntax structure starts to be assembled on the bases of the grammatical information on the input of Z_{27}. The edge: *{X, Verb}* entails[1] a Semantic Merge between X* and V

As supposed in the argument-based syntactic model, the subject starts first the treatment.

$$M (X^*V^*) = [X^*, V]\ (\underline{?O}, ?R) \tag{7}$$

The result of M consists of a couple, in which each element obtains an image μ, representing the concept in the semantic context of the other member of the couple. For example, the image μ of the concept X* within the couple [X*V] is the image of X* as Subject, performing V:

$$\mu\ X^* \in [X^*V] = X^*\ \text{Acts}\ (\underline{?O}, R) \tag{8}$$

The other grammatical edge on the input is {*на*, Z}.

Preposition *на* indicates that the noun that follows it is either Recipient, or Possessor.

The final syntactic result depends on the decision upon the meaning of *на*. Following the experimental data (Table 3), the final syntactic result is not assigned depending on *на.*

As it is seen in table 3, *The Boy* is totally rejected as a Possessor of *the Apples* without any reason residing within the couple *Boy-Apples*. In fact, all the used Z* can possess Y* in a specific context. The only possible interpretation of this experimental result is that the meaning of *на* is assigned later, depending on the obtained images μ of X*, Y* and Z* within the action V*. That means that, as suggested in AGN model, the word-forms and their semantic images stay available in working memory until the end of the syntactic treatment.

---

[1] The word order in Bulgarian is flexible and the fact that the statements are in canonic word-order S-V-O is not a sufficient condition to assign to X the role of Subject.

The grammatical information brought to the input of *Parse* is not sufficient to build a syntactic structure. The treatment can continue only by applying other mechanisms.

The schemes on figures 5 and 6 represent the supposed steps of syntactic treatment with Semantic Merge for the Recipient assignment and the Possessor assignment. The images μ obtained at each step are given in order to figure out the mechanism of the treatment and to analyze it in comparison with the experimental results.

Following the experiment, the result of one of the treatments is **rejected**. The assumption is that the intermediate images μ of either the Recipient scheme (figure 5) or the Possessor scheme (figure 6) are rejected by activating the semantic relations preexisting in NSET.

The temporal semantic result of step 2 of each scheme is:

$$\mu Z^* \in [\mu X^*, Z^*] = \text{Recipient}(\mu X^*) \tag{9}$$

$$\mu Z^* = (Z^* \leftarrow Y^*) = \text{Possessor of } Y^* \tag{10}$$

The experimental results in Table 3 show that the treatment follows the Recipient scheme (fig. 5) even when there are no reasons to reject μZ^* as Possessor of Y*. The explanation of this result is that **the argument-based syntactic structure is "calculated" as primary**. If there are no reasons to reject it, it is accepted as final. That confirms the assumption that the argument-based syntax has a fundamental character.

Table 6. Influence of the Possessor relation $Z^* \leftarrow Y^*$

μ X*		Z*	Y*	to%
Anna	Sold	the boy	The apples	100
Mihail	Sold	His neighbor	The house	71
Ivan	Showed	His father	The path	100
Peter	Showed	His father	The house	67

However, as the experimental results in table 6 show, if in the semantic space there exists a previous knowledge about Z* as a natural Possessor of Y*, that influences the process in step 2. The preexisting relations of Possessor $Z^* \leftarrow Y^*$ between *father* ← *house* and *neighbor*← *house* influence the final decision which starts switching to the Possessor decision.

Further syntactic treatment must assign to the merged couple an Object (?O), because the verbs used are transitive, and the statements necessarily must have an Object.

In the Recipient scheme (fig. 5) this is made in step 3, where a Merge is performed between μX^* and Y*. The result is $[\mu X^*, Y^*]$ where Y* is the Object of μX^*. In step 4 the treatment terminates by merging $[\mu X^*, Y^*]$ and $[\mu X^*, Z^*]$.

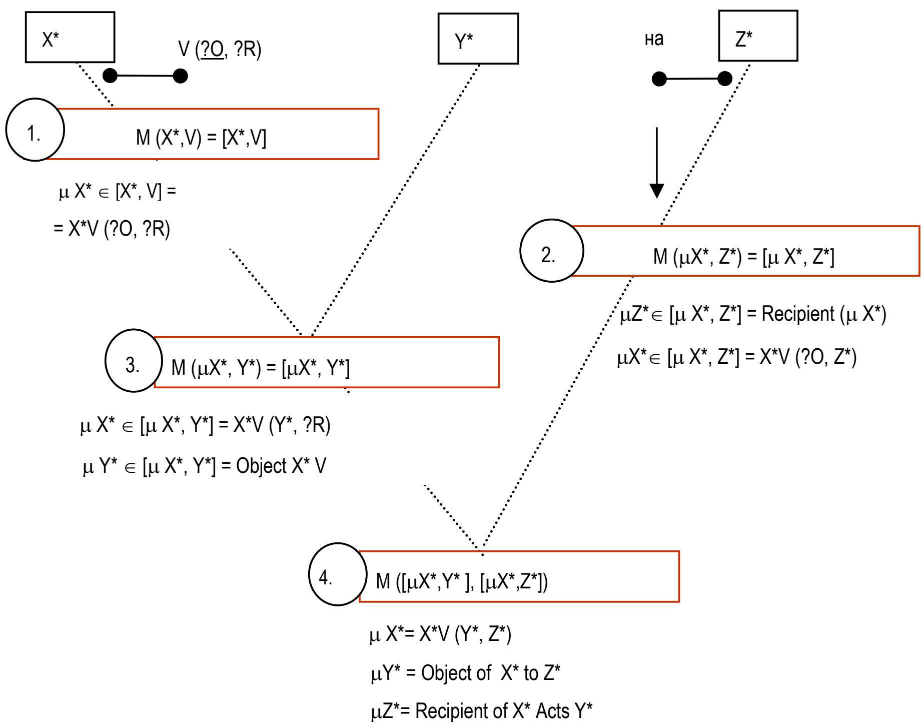


Fig. 5. Recipient scheme

In the Possessor scheme (fig. 6), Z* is assigned Possessor of Y* in step 2a. In step 2b the treatment "knows that Z* is busy" and assigns Ø to the Recipient position of μX*. In step 3 μY*=(Y*→Z*) is assigned as Object of μX*. The result [μX*, μY*] is merged in step 4 with [μX*, Ø].

The assignment of Object is influenced by the relations existing in the semantic space. The knowledge about Y* as the usual Object of actions of X* forms in the semantic space a relation X*-Y*. This relation matches the result of syntactic treatment when merging μX* and Y*. Such is the case with the couples *Fitter–Car* and *Upholster-Chair* in the examples:

The pre-existing relation X*-Y* "pushes" the treatment to Merge μX* and Y*directly, as it is in the Possessor scheme. In terms of AGN semantic activation, the concept of *the Fitter* and of *the Car* will be activated as images of Wi and as related to each other nodes. The same is with *the Upholster* and *the Chair*.

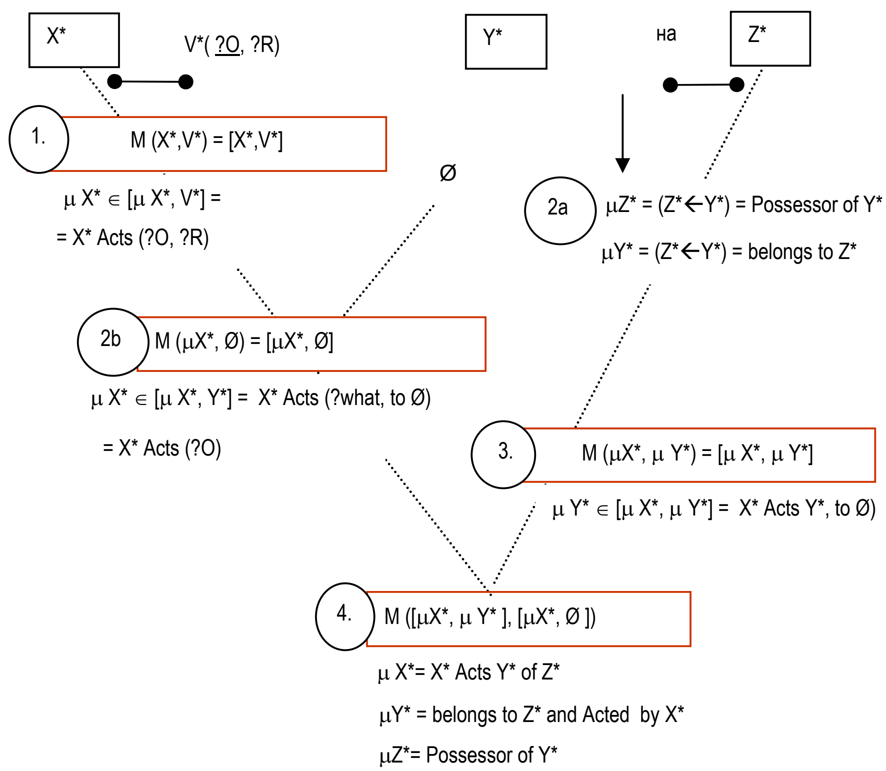


Fig. 6. Possessor scheme

The pre-existing semantic relation X*-Z* also influences the treatment. As seen in Table 7, *The Dog* is rejected as Recipient of *Elena Sold* and accepted as Recipient of *The Big Bad Wolf Sold*. *The Dog* is not rejected as Recipient of *Sold* in general. In terms of semantic activation of concepts and features, the concept of *The Big Bad Wolf* activates animals as possible Recipients of the actions. This example shows that when between X* and Z* there exists a strong semantic relation, Z* takes the (natural) role of Recipient of the actions of X*.

Table 7. Influence of the relation X*-Y*

X*	Y*			Z*	To%	Assignment
Mary	The car	Showed		the neighbor	89	Recipient
The fitter	The car	Showed		the neighbor	50	Equivalence
Peter	The chair	Brought		The director	88	Recipient
The upholster	The chair	Brought		The director	50	Equivalence

Table 8. Influence of the relation X*-Z*

μX*		Z*		Y*	Result	Assignment
Elena	Sold	The dog		The house	100% of	Possessor
The Big Bad Wolf	Sold	The dog		The house	67% to	Recipient

The existing semantic relations between all the three of the concepts X* Y* and Z* influence the syntactic treatment in the example below, where the Recipient result is rejected in the second sentence and it has a Possessor interpretation.

Table 9.

μX*		Y*	Z*	Of%	To%	Assignment
Mihail	Sold	The house	His neighbor	29	71	Recipient
Ivan	Sold	The house	His father	67	33	Possessor

The pre-existing semantic Possessor relation Z*←Y*: *Father ← house* pushes the treatment to the Possessor tendency. Following the result in table 6, that influence is less strong than as observed in this example. The existing strong relation X*-Z*: *Ivan-his Father* entails a Recipient tendency (table 8) and there are no reasons to reject [μX* Z*] *IvanSold-his Father.* The experimental result could be explained with the existence of a strong relation X*-Y*: *Ivan – the house* which influences the treatment (table 7) in addition of *Father← house*. Such a strong relation X*-Y* could exist if:

(X*-Z*) And (Z* ← Y*) ⇒ (X*-Y*)

(*Ivan - his Father*) And (*his Father ← House*) ⇒ (*Ivan – house*)

In the semantic space "Ivan usually operates with his father's property".

The general conclusion is that people construct the meaning of what has been said.

Conclusions and Future Work

The general suggestion in our approach is that interactions exist between the purely language features and some semantic fundamental which is similar for all languages.

Assumptions about how the argument structure is computed have led to the development of the argument-based model of basic syntax. The reiterative operation assigns a primary role to entities as the key components of syntactic structure.

Experimental evidence is provided in support of the argument-based model. The semantic role of entities (nouns) seems to be primary in syntax. The experimental data show that mental computation of syntax is influenced by the inter-conceptual relations between the images of entities in a semantic space. The analysis provided here is based on the assumption that the syntactic treatment includes a Merge operation between the images of the concepts. The formal description obtained of the stages of syntactic treatment corresponds to the experimental results. That supports the hypothesis that the grammatical rules entail operations on semantic primitives.

Further study will require a more precise picture of the dependencies between semantic primitives, lexical items, and syntactic rules. That will lead to an advanced modeling of the phenomenon under examination.

Bibliography

[Carnie, 2006] Carnie, Andrew, (2006) Syntax: A Generative Introduction. Blackwell.

[Chomsky, 2004] The Generative Enterprise Revisited. Mouton de Gruyter.

[Chomsky, 2006] Noam Chomsky. Biolinguistic Explorations: Design, Development, Evolution, 2006, West Hall, Bathish Auditorium, AUB.

[Hauser et al., 2002] M. Hauser, N. Chomsky and W.T. Fitch. The Faculty of Language: What is it, who has it, and how did it evolve? In: Science Vol. 298.

[Slavova, 2004] Slavova, Velina (2004) A generalized net for natural language comprehension. In: Advanced Studies in Contemporary Mathematics, vol 8, Ku-Duk Press, 131-153.

[Slavova, Soschen, 2007] Slavova, V., and A. Soschen (2007), A Fibonacci-tree model of cognitive processes underlying language faculty, in: Proc. Of 3-rd international conference in Computer Science, NBU, University of Fulda, Boston University, pp. 196-205.

[Slavova, Soschen, Immes, 2005] Slavova, V., Soschen A. and L. Immes, (2005), Information processing in a cognitive model of NLP, in: International Journal Information theories & applications, vol 12, N3, pp 157 – 166

[Slavova, Soschen, 2008] [Soschen, Slavova 2007] Slavova V., and A. Soschen (2008), Experimental support of argument-based syntactic computation, in: proc. of ITA 2008, Joint International scientific Events on Informatics, Varna 2008.

[Soschen, 2005] Soschen, Alona. 2005. Derivation by phase: Russian Applicatives. Canadian Linguistic Association Conference proceedings.

[Soschen, 2006] Soschen, Alona (2006). Natural Law and the Dynamics of Syntax (MP). Linguistics in Potsdam 25. Optimality Theory and Minimalism: a Possible Convergence? Hans Broekhius and Ralf Vogel (eds.): ZAS, Berlin.

[Soschen, 2008] Soschen, Alona (2008). On the Nature of Syntax. Bio-linguistics Journal, V.2/2.

[Soschen, Slavova, 2007] Soschen A., and V. Slavova (2007), Cognitive modeling of recursive mechanisms in syntactic processing. In: proc. of the IX international conference in Cognitive Modeling n Linguistics, Text processing and cognitive linguistics, pp 334-343

Authors' Information

Velina Slavova *- New Bulgarian University, Department of Computer Science, 21 Montevideo str., 1618 Sofia, Bulgaria, e-mail: vslavova@nbu.bg*

Major Fields of Scientific Research: Cognitive modeling in linguistics, formal models of language phenomena

Alona Soschen *- Massachusetts Institute of Technology, MIT Department of Linguistics and Philosophy, 77 Massachusetts Ave. 32-D808Cambridge, MA 02139-4307, USA, e-mail: soschen@mit.edu*

Major Fields of Scientific Research general features of biological systems present in linguistic structures

ADAPTIVE COMPARTMENTAL WAVELON WITH ROBUST LEARNING ALGORITHM

Yevgeniy Bodyanskiy, Oleksandr Pavlov, Olena Vynokurova

Abstract*: In this paper a robust learning algorithm for adaptive compartmental wavelon based on R. Welsh criterion is proposed. Suggested learning algorithm under consideration allows the signals processing in presence of significant noise level and outliers. The robust learning algorithm efficiency is investigated and confirmed by the number of experiments including bio-medical applications.*

Keywords*: computational intelligence, hybrid architecture, wavelet, adaptive compartmental wavelon, robust learning algorithm, outliers resistant.*

ACM Classification Keywords*: I.2.6 Learning – Connectionism and neural nets.*

Introduction

Nowadays artificial neural networks (ANN) have gained the significant prevalence for solving the wide class of the information processing problems, uppermost for the identification, emulation, intelligent control, time series forecasting of arbitrary kind under significant noise level, and also the structural and parametric uncertainty.

The multilayer feedforward networks of three-layer perceptron type, where the elementary nodes are so-called P - neurons with monotonic activation functions are the most known and popular. The efficiency of the multilayer networks is explained by their universal approximation properties in combination with relative compact presentation of the simulated nonlinear system. It means, that they can be used successfully in the simulation (emulation) of non-linear systems tasks, which can be described by the equation

$$y(k) = F(x(k)) + \xi(k), \tag{1}$$

where $y(k)$ is the output system signal in k -th instant of discrete time $k = 0,1,2,\ldots$, $x(k) \in X$ - $(n \times 1)$ is the vector of input signal, including both exogenous variables and previous values of the output signal, $F(\bullet)$ is the arbitrary function, generally in some unknown form, $\xi(k)$ is the unobserved disturbance with unknown characteristics. Usually it is assumed that function $F(\bullet)$ is defined either on the unit hypercube or on the orthotop

$$x_i(k) \in [x_i^{\min}, x_i^{\max}],\ i = 1,2,\ldots,n,$$

where $x_i^{\min}$, $x_i^{\max}$ are the known low and upper limits of the i -th input influence variation.

The principal disadvantage of the multilayer networks is the low learning rate which is based on backpropagation algorithm which makes their application in the real time tasks impossible.

Alternative to the multilayer ANNs are the radial basis function networks, having one hidden layer consisting of, so-called, R -neurons. These networks learning is realized on the level of the output layer which is usually represented by the adaptive linear associator [Moody, Darken, 1989; Moody, Darken, 1988; Park, Sandberg, 1991; Leonard, 1992; Sunil, Yung, 1994; Poggio, Girosi, 1994]. Unlike P-neurons, R-neurons conventionally have bell-shaped activation function $f_j(x)$, where the argument is a distance (usually in Euclidean metric) between the current value of input signal $x(k)$ and the center c_j of the j-th neuron, i.e.

$$\varphi_j(x(k)) = \varphi_j\left(\sum_{i=1}^{n}(x_i(k) - c_{ji})^2\right) = \varphi_j\left(\left\|x(k) - c_j\right\|^2\right) \qquad (2)$$

The principal advantage of RBFN is the high learning rate in the output layer, because the tuning parameters are linearly included to the network description. At the same time the problem of R-neurons centers allocation is remaining, and its unsuccessful solving leads to the «curse of dimensionality» problem. Using clustering techniques though allows reducing the size of the network, but excludes the possibility of on-line operation. Here it can be noted, that in [Bishop, 1995] the gradient recurrent procedure of the component-wise tuning parameters c_{ji} is described, but it is characterized by the low learning rate.

Along with neural networks for the arbitrary type signals processing, in the last years the wavelet theory is used sufficiently often [Chui, 1992; Daubechies, 1992], providing the compact local signal presentation both in the frequency and time domains. At the turn of the artificial neural network and wavelets theories the wavelet neural networks [Billings, 2005; Zhang, Benveniste, 1992; Zhang, 1997; Bodyanskiy, Vynokurova et al, 2006; Bodyanskiy, Vynokurova et al, 2007; Bodyanskiy et al, 2007; Bodyanskiy, Vynokurova et al, 2008] have evolved their efficiency for the analysis of non-stationary nonlinear signals and processes.

Elementary nodes of the wavelet neural networks are so-called radial wavelons [Reyneri, 1999], where the activation functions are the even wavelets with argument in form the Euclidian distance between $x(k)$ and wavelet translation vector c_j, where that every component of distance $\left|x_i(k) - c_{ji}\right|$ is weighted by the dilation parameter σ_{ji} such, that

$$\varphi_j(x(k)) = \varphi_j\left(\sum_{i=1}^{n}\left((x_i(k) - c_{ji})/\sigma_{ji}\right)^2\right) \qquad (3)$$

where $\varphi_j(\bullet)$ is wavelet activation function. The receptive fields for such wavelons are hyperellipsoids with axes which are collinear to coordinate axes of the space X.

Taking into consideration the equivalence of radial basis ANN and fuzzy inference systems [Jang, Sun, 1993; Hunt et al, 1996], and also possibility of using even wavelet as a membership function [Mitaim, Kosko, 1996; Mitaim, Kosko 1997], within the bounds of the unification paradigm [Reyneri, 1999] we can talk about such hybrid system as adaptive compartmental wavelon having the radial-basis function network fast learning ability, fuzzy inferences systems interpretability and wavelet's local properties.

It can be noted, that mostly tuning algorithms based on traditional squared learning criteria in the case of the processing data being contaminated by outliers with unknown distribution law, have shown themselves very sensitive to anomalous outliers. Thus the actual task is a synthesis of the robust learning algorithms, thet allow signal processing in presence of anomalous outliers.

This paper is devoted to synthesis of robust learning algorithm for adaptive compartmental wavelon, which has adjustable level of insensitivity to the different kind of outliers, rough errors, non-Gaussian disturbances, has high convergence rate and provides the advanced approximation properties in comparison with conventional computational intelligence systems. This hybrid structure can be used as the nodes of the neural network architectures.

1. Adaptive Compartmental Wavelon Structure

Let us consider the two-layers architecture shown on fig. 1 that coincides with the traditional radial-basis neural network. The input layer of the architecture is the receptor and in current time instant k the input signal in vector form $x(k)=(x_1(k), x_2(k), \dots, x_n(k))^T$ is fed on it. Unlike radial basis function network the hidden layer consists of not by R-neurons, but by wavelons with wavelet activation function in the form

$$\varphi_j(x(k)) = \varphi_j\left(\left((x(k)-c_j)^T Q_j^{-1}(x(k)-c_j)\right), \alpha_j\right), \quad j=1,2,\dots,h \qquad (4)$$

in which instead of translation parameters σ_{ji} in (1) the dilation matrix Q_j is used, i.e. it is not Euclidian distance, but Itakura-Saito metric [Itakura, 1975].

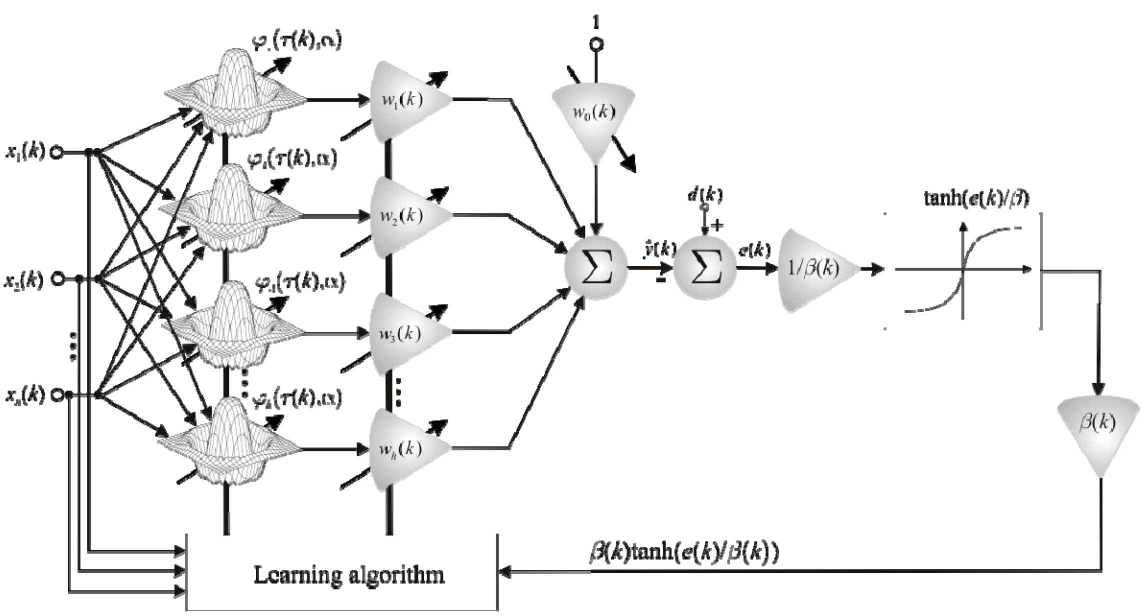


Fig. 1 – Adaptive compartmental wavelon with robust block

This results to the fact that receptive fields – wavelons hyperellipsoids (2) can have the arbitrary orientation relatively to the coordinate axes of space X, what extends the functional properties of adaptive compartmental wavelon.

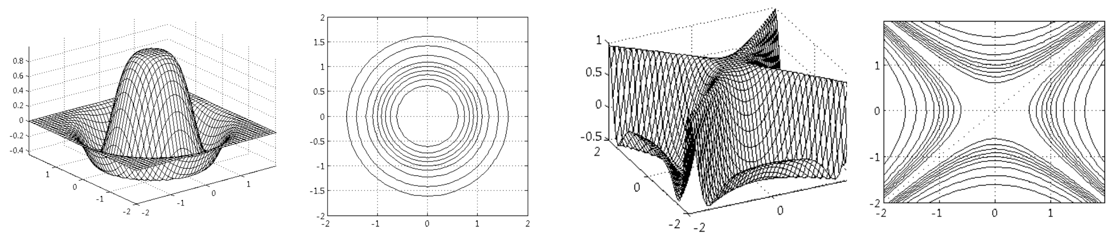
Based on the results of the author [Mitaim, Kosko, 1996; Mitaim, Kosko 1997], about that the wavelet-function can be used as a membership function in fuzzy systems, we can introduce the adaptive membership function based on wavelet Mexican Hat, having form

$$\varphi_j(\tau_j(k)) = (1-\alpha_j \tau_j^{\,2}(x(k)))\exp\left(-\frac{\tau_j^{\,2}(x(k))}{2}\right) \quad (5)$$

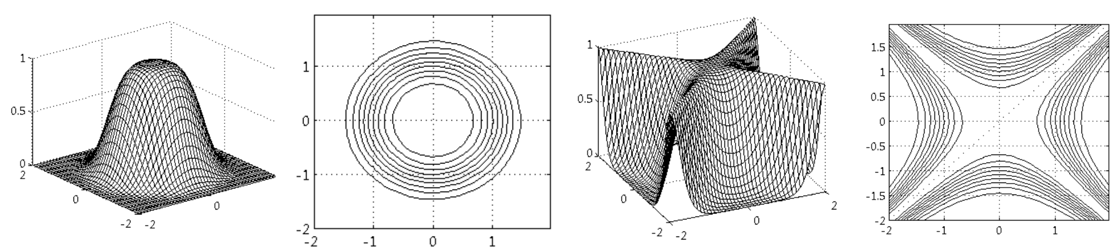
where $\tau_j(x(k)) = \left((x(k)-c_j(k))^T Q_j^{-1}(k)(x(k)-c_j(k))\right)$, α_j is turning parameter ($0 \le \alpha \le 1$).

Adaptive parameter α_j allows to tune the form of membership function in process of hybrid architecture learning, thus if $\alpha = 0$ then we get Gauss membership function, if $\alpha = 1$ then we get wavelet membership function Mexican Hat, and if $0 < \alpha < 1$ then we get hybrid membership function.

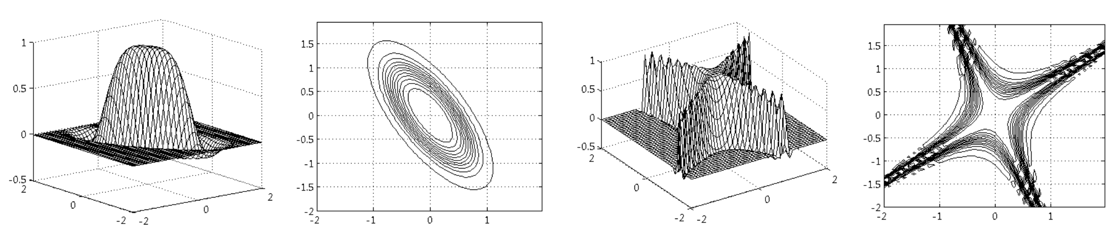
Fig. 2 shows the wavelon activation function (5) with arbitrary matrices Q_j and parameter α_j .



a) $Q^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \alpha = 1$ b) $Q^{-1} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \alpha = 1$



c) $Q^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \alpha = 0$ d) $Q^{-1} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \alpha = 0$



f) $Q^{-1} = \begin{pmatrix} 4 & 3 \\ 1 & 2 \end{pmatrix}, \alpha = 0.5$ g) $Q^{-1} = \begin{pmatrix} -4 & 3 \\ 1 & 2 \end{pmatrix}, \alpha = 0.5$

Fig. 2 – Wavelon activation function with arbitrary matrices Q_j and parameter α_j

And at last, the output layer is the common adaptive linear associator with tuning synaptic weights w_j

$$\hat{y}(k) = w_0(k) + \sum_{j=1}^{h} w_j(k)\varphi\Big(\big((x(k)-c_j)^T Q_j^{-1}(x(k)-c_j)\big), \alpha_j(k)\Big) = w(k)^T \varphi(\tau(k), \alpha), \quad (6)$$

where $\varphi_0(\tau(k)) \equiv 1$, $w(k) = (w_0(k), w_1(k), w_2(k), \ldots, w_h(k))^T$, $\varphi(\tau(k), \alpha) = (1, \varphi_1(\tau_1(k), \alpha_1), \varphi_2(\tau_2(k), \alpha_2),$ $\ldots, \varphi_h(\tau_h(k), \alpha_h))^T$, $\tau(k) = (x(k)-c_j)^T Q_j^{-1}(x(k)-c_j)$.

Thus the tuning parameters of architecture to be determined in the learning process form the set of the $h+1$ synaptic weights w_j, h $(n \times 1)$ -vectors c_j, h $(n \times n)$ -matrices Q_j^{-1} and h $(n \times 1)$ -vectors α_j. In total such network includes h $(1 + 2n + n^2) + 1$ adjustable parameters.

2. The Robust Learning Algorithm for Adaptive Compartmental Wavelon

The experience shows that the identification methods based on the least square criterion are extremely sensitive to the deviation of real data distribution law from Gaussian distribution. In presence of various type outliers, an outrage errors, and non-Gaussian disturbances with “heavy tails” the methods based on the least squares criterion loose their efficiency.

In this case the methods of robust estimation and identification [Rey, 1978] which have obtained the wide spread for the learning of the artificial neural networks [Cichocki, Lobos, 1992; Cichocki, Unbehauen, 1993; Li, Chen, 2002] appear on the first role.

Let’s introduce into the consideration the learning error

$$e(k) = y(k) - \hat{y}(k) = y(k) - w^T(k)\varphi(k) \quad (7)$$

and robust identification criterion by R. Welsh [Holland, Welsh, 1977; Welsh, 1977]

$$E(k) = f(k) = \beta^2 \ln\big(\cosh\big(e(k)/\beta\big)\big) \quad (8)$$

where β is a positive parameter, that is chosen from empirical reasons and defining the size of zone of tolerance to outliers. It is necessary to note, that robust criterion (8) satisfies to all metric space axioms.

Fig. 3 shows the comparison of the robust optimization criterion with the least squares criterion.

Further we shall consider the learning algorithms synthesis. For the synaptic weights and the waveleon parameters (vectors c_j and matrices Q_j^{-1}) tuning we use gradient minimization of criterion (9), thus unlike the component-wise learning considered in [Bishop, 1995], we make some corrections in the vector-matrix form, that, firstly is easier from computing point of view, and secondly it allows to optimize learning process on the operation rate.

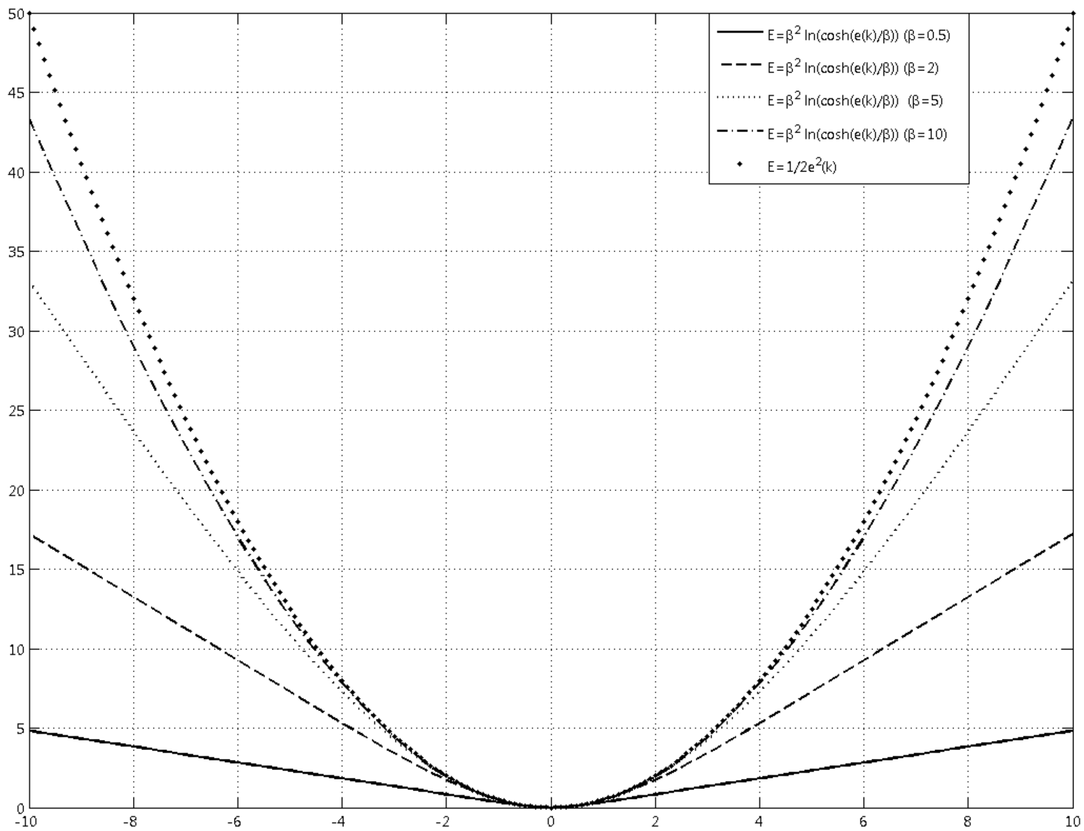


Fig. 3 – Comparison of the optimization criteria

In general case the learning algorithm can be written in form

$$\begin{cases} w(k+1) = w(k) - \eta_w \nabla_w E(k), \\ c_j(k+1) = c_j(k) - \eta_{c_j} \nabla_{c_j} E(k), \;\; j = 1,2,\ldots,h, \\ Q_j^{-1}(k+1) = Q_j^{-1}(k) - \eta_{Q_j^{-1}} \left\{ \partial E(k) \big/ \partial Q_j^{-1} \right\}, \;\; j = 1,2,\ldots,h, \\ \alpha(k+1) = \alpha(k) - \eta_\alpha \nabla_\alpha E(k), \end{cases} \qquad (9)$$

where $\nabla_w E$ is vector-gradient of the criterion (8) on w, $\nabla_{c_j} E$ is $(n \times 1)$ -vector-gradient criterion (9) on c_j; $\left\{ \partial E(k) \big/ \partial Q_j^{-1} \right\}$ is $(n \times n)$ -matrix, formed by partial derivatives $E(k)$ on components Q_j^{-1}; $\nabla_\alpha E$ is $(n \times 1)$ -vector-gradient criterion (9) on α; η_w, η_{c_j}, $\eta_{Q_j^{-1}}$, η_α are the learning rates.

For the wavelet (5) we can write

$$\begin{cases} \nabla_w E(k) = -\beta \tanh\left(e(k)/\beta\right)(1-\alpha_j \tau_j^{\ 2}(x(k)))\exp\left(-\frac{\tau_j^{\ 2}(x(k))}{2}\right) = -\tanh\left(e(k)/\beta\right) J_w(k), \\ \nabla_{c_j} E(k) = \beta \tanh\left(e(k)/\beta\right) w_j(k)\left(\alpha\tau_j^3(x(k)) - (2\alpha+1)\tau_j(x(k))\right)\cdot \\ \cdot\exp\left(-\frac{\tau_j^2(x(k))}{2}\right) Q_j^{-1}(k)(x(k)-c_j(k)) = \tanh\left(e(k)/\beta\right) J_{c_j}(k), \\ \left\{\partial E(k)/\partial Q_j^{-1}\right\} = -\beta \tanh\left(e(k)/\beta\right) w_j(k)\left(\alpha\tau_j^3(x(k)) - (2\alpha+1)\tau_j(x(k))\right)\cdot \\ \cdot\exp\left(-\frac{\tau_j^2(x(k))}{2}\right)(x(k)-c_j(k))(x(k)-c_j(k))^T = -\tanh\left(e(k)/\beta\right) J_{Q_j^{-1}}(k), \\ \nabla_\alpha E(k) = -\beta \tanh\left(e(k)/\beta\right) w(k)\tau^2(x(k))\exp\left(-\frac{\tau^2(x(k))}{2}\right) = \tanh\left(e(k)/\beta\right) J_\alpha(k), \end{cases} \qquad (10)$$

where $\tau_j(x(k)) = (x(k)-c_j(k))^T Q_j^{-1}(k)(x(k)-c_j(k))$.

Then the wavelons learning algorithm of the hidden layer subject to (10) is taking the form

$$\begin{cases} w(k+1) = w(k) + \eta_w \tanh\left(e(k)/\beta\right) J_w(k), \\ c_j(k+1) = c_j(k) - \eta_{c_j} \tanh\left(e(k)/\beta\right) J_{c_j}(k), \\ Q_j^{-1}(k+1) = Q_j^{-1}(k) + \eta_{Q_j^{-1}} \tanh\left(e(k)/\beta\right) J_{Q_j^{-1}}(k), \\ \alpha(k+1) = \alpha(k) + \eta_\alpha \tanh\left(e(k)/\beta\right) J_\alpha(k), \end{cases} \qquad (11)$$

at that convergence rate to the optimal value w, c_j, Q_j^{-1} and α is completely defined by learning rate parameters η_w, η_{c_j}, $\eta_{Q_j^{-1}}$, and η_α.

The learning rate increasing can be achieved by using procedures more complex than gradient ones, such as Hartley or Marquardt procedures, that for the first relation (11) can be written in general form [Bodyanskiy, 1987; Bodyanskiy, Vynokurova et al, 2008]

$$w(k+1) = w(k) - \lambda_w (J_w(k) J_w^T(k) + \eta_w I)^{-1} J_w(k) \tanh\left(e(k)/\beta\right), \qquad (12)$$

where I is the $(n\times n)$-identity matrix, λ_w is a positive dampening parameter, η_w is a momentum term parameter.

Using the inverse matrices lemma and after applying simple transformations we obtain the effective parameters learning algorithm in the form

$$\begin{cases} w(k+1) = w(k) + \lambda_w \left(\tanh\left(e(k)/\beta\right) J_w(k)\right) \Big/ \left(\eta_w + \left\|J_w(k)\right\|^2\right), \\ c_j(k+1) = c_j(k) - \lambda_c \left(\tanh\left(e(k)/\beta\right) J_{c_j}(k)\right) \Big/ \left(\eta_c + \left\|J_{c_j}(k)\right\|^2\right), \\ Q_j^{-1}(k+1) = Q_j^{-1}(k) + \lambda_{Q_j^{-1}} \left(\tanh\left(e(k)/\beta\right) J_{Q_j^{-1}}(k)\right) \Big/ \left(\eta_{Q_j^{-1}} + Tr(J^T_{Q_j^{-1}}(k) J_{Q_j^{-1}}(k))\right), \\ \alpha(k+1) = \alpha(k) + \lambda_\alpha \left(\tanh\left(e(k)/\beta\right) J_\alpha(k)\right) \Big/ \left(\eta_\alpha + \left\|J_\alpha(k)\right\|^2\right). \end{cases} \tag{13}$$

In order to add more smoothing properties, using approach proposed in [Bodyanskiy et al, 2001; Bodyanskiy, Vynokurova et al, 2008], we can introduce the modified learning procedure:

$$\begin{cases} w(k+1) = w(k) + \lambda_w \dfrac{\tanh\left(e(k)/\beta\right) J_w(k)}{\eta_w(k)}, & \eta_w(k+1) = \gamma\eta_w(k) + \left\|J_w(k+1)\right\|^2, \\ c_j(k+1) = c_j(k) - \lambda_{c_j} \dfrac{\tanh\left(e(k)/\beta\right) J_{c_j}(k)}{\eta_{c_j}(k)}, & \eta_{c_j}(k+1) = \alpha_c \eta_{c_j}(k) + \left\|J_{c_j}(k+1)\right\|^2, \\ Q_j^{-1}(k+1) = Q_j^{-1}(k) + \lambda_{Q_j^{-1}} \dfrac{\tanh\left(e(k)/\beta\right) J_{Q_j^{-1}}(k)}{\eta_{Q_j^{-1}}(k)}, & \eta_{Q_j^{-1}}(k+1) = \gamma\eta_{Q_j^{-1}}(k) + Tr\left(J^T_{Q_j^{-1}}(k+1) J_{Q_j^{-1}}(k+1)\right) \\ \alpha(k+1) = \alpha(k) + \lambda_\alpha \dfrac{\tanh\left(e(k)/\beta\right) J_\alpha(k)}{\eta_\alpha(k)}, & \eta_\alpha(k+1) = \gamma\eta_\alpha(k) + \left\|J_\alpha(k+1)\right\|^2, \end{cases} \tag{14}$$

(here $0 \le \gamma \le 1$ are the parameters of weighting out-dated information), being nonlinear hybrid of the Kaczmarz-Widrow-Hoff and Goodwin-Ramadge-Caines algorithms and including both following and filtering properties.

3. Results of the Experimental Research

In the first experiment the developed robust learning algorithm was tested out on the basis of a signal with intensive outliers. The signal had been obtained using Narendra's nonlinear dynamical system (it is a standard benchmark, widely used to evaluate and compare the performance of neural and neuro-fuzzy systems for nonlinear system modeling and time series forecasting) whose output signal is artificially contaminated by random noise generated according to the Cauchy distribution with the inverse transform method described by equation in form

$$F_X^{-1}(x) = x_0 + \gamma tg\left[\pi\left(x - 0.5\right)\right] \tag{15}$$

where x_0 is the location parameter, γ is the scale parameter $(\gamma > 0)$, x is the support area ($x \in (-\infty, +\infty)$).

The nonlinear dynamical system is generated by equation in form [Narendra, Parthasarathy, 1990]

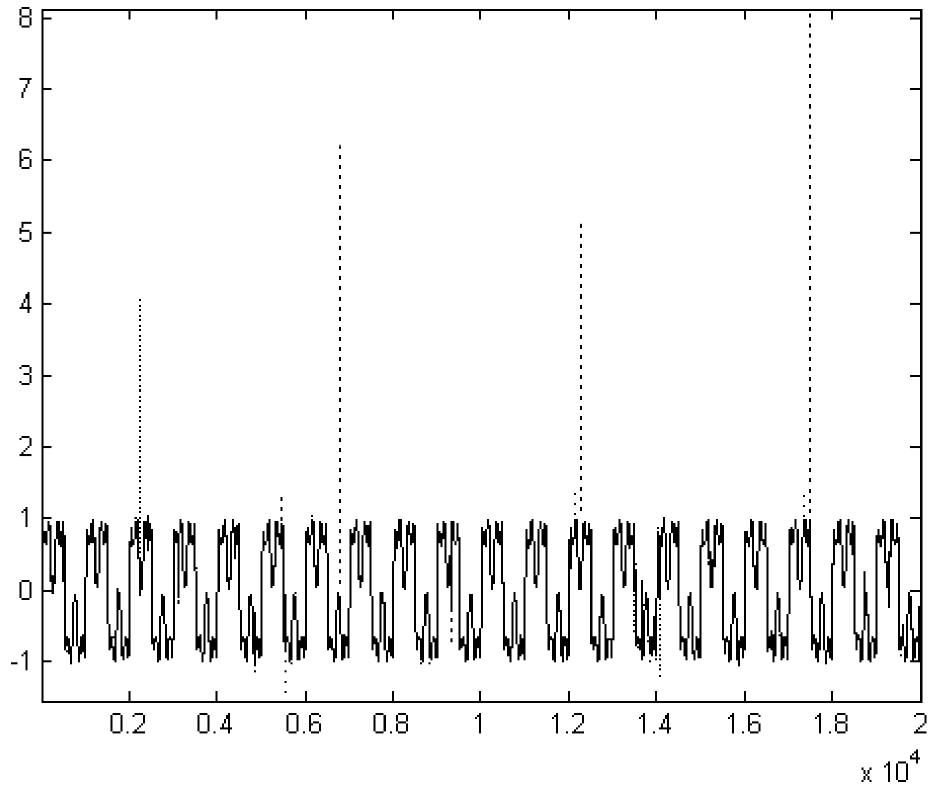
$$y(k+1) = 0.3y(k) + 0.6y(k-1) + f(u(k)), \tag{16}$$

where $f(u(k)) = 0.6\sin(\pi u(k)) + 0.3\sin(3\pi u(k)) + 0.1\sin(5\pi u(k))$ and $u(k) = \sin(2\pi k/250)$, $0 < k < 20000$ is discrete time. The values $x(t-2), x(t-1), x(t)$ were used to emulate $x(t+1)$. In the on-line mode of learning, adaptive compartmental wavelon was trained with procedure (13) for 20000 iterations. The parameters of the learning algorithm were $\beta_w = 0.5, \beta_c = 0.5, \beta_Q = 0.5, \beta_\alpha = 0.5,$ $\gamma_w = \gamma_c = \gamma_Q = \gamma_\alpha = 0.99,$ $\lambda_w = \lambda_{c_j} = \lambda_{Q_j^{-1}} = \lambda_\alpha = 0.99$. Initial values were $\eta_w(0) = \eta_{c_j}(0) = \eta_{Q_j^{-1}}(0) = \eta_\alpha(0) = 10000$. After 20000 iterations the training was stopped, and the next 5000 points were used as the testing data set. Testing data set included $f(u(k)) = 0.6\sin(\pi u(k)) + 0.3\sin(3\pi u(k)) + 0.1\sin(5\pi u(k))$ and $u(k) = \sin(2\pi k/250)$ for $20001 < k < 22000$ and $f(u(k)) = u^3(k) + 0.3u^2(k) - 0.4u(k)$ and $u(k) = \sin(2\pi k/250) + \sin(2\pi k/25)$ for $22001 < k < 25000$.

Fig. 3 a shows the results of the noised nonstationary signal emulation (real values (dashed line) and emulated values (solid line)). Fig. 3 b shows segment of the learning process; as it can be seen the number of outliers with large amplitude, present in the beginning of the sample, didn't have a significant influence on the learning algorithm.

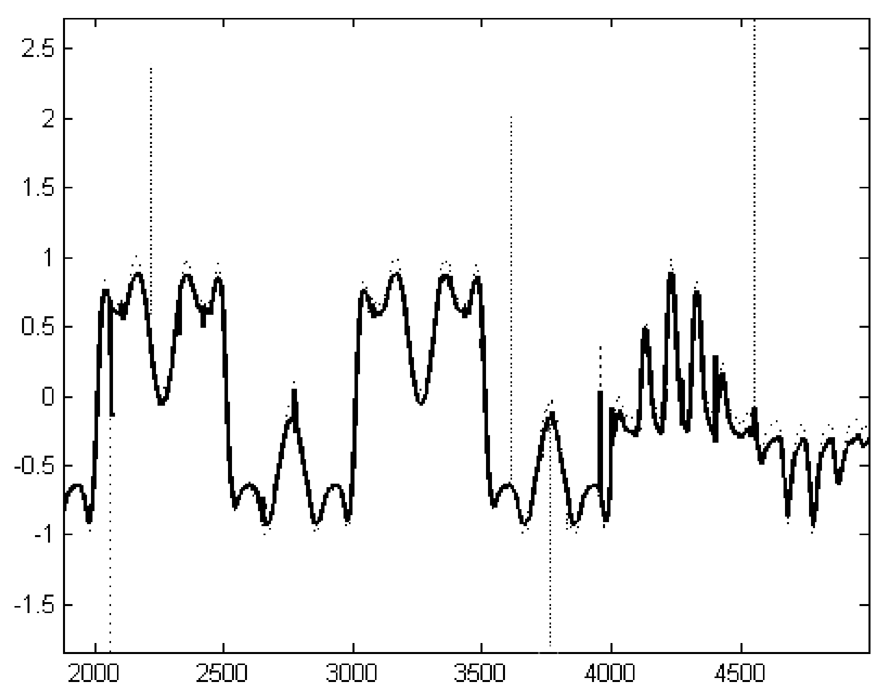
The comparison of emulation results based on robust learning algorithm with results of emulation based on gradient algorithm and the algorithm based on recurrent least squares method where the structure network and the number of tuning parameters were identical was carried out.

Under adaptive compartmental wavelon learning using the gradient algorithm the first outlier in the beginning of the sample, had a noticeable influence on the learning algorithm. Under adaptive compartmental wavelon learning using the recurrent least-squares method the first occurred outlier leads to the covariance matrix so-called "parameters blow-up" what results in inability to emulate signals noised by anomalous outliers. Thus it is obvious that the proposed robust learning algorithm allows signal processing under high level outliers noise conditions.



a)

RMSEchk=0.0657



b)

Fig. 3 – Results of noised nonstationary signal emulation based on robust learning algorithm

In Table 1 the comparison results are shown.

Table 1: The results of noisy signal emulation

Neural Network / Learning algorithm	RMSE
Adaptive compartmental wavelon / Proposed robust learning algorithm (13) with parameter α	0.0657
Adaptive compartmental wavelon / Proposed robust learning algorithm (13) without parameter α	0.0998
Adaptive compartmental wavelon / The gradient learning algorithm	1.1436
Adaptive compartmental wavelon / RLSM	∞

The second experiment has been made on the data set, presented by Government Institution "Institute of General and Urgent Surgery (Academy of Medical Sciences of Ukraine)". It has been carried out studying of the homeostasis indexes dynamic of the patient with the stomach acute injury [Cook et al, 1996; Vynokurova, Pavlov et al, 2007] based on outliers resistant radial-basis-fuzzy-wavelet-neural network. The indexes of oxygen cascade, system hemodynamics, daily pH- measurement, and hypoxia marker and endotoxemia were analyzed. Result of processing studied clinico-laboratory data set was the degree defining of enteral deficiency, that it has allowed to lead the adequate stomach-protect diagnosis and therapy.

Conclusion

In the paper computationally simple and effective all adaptive compartmental wavelon parameters robust learning algorithm is proposed. The robust learning algorithm has following and smoothing properties and allows on-line processing of nonstationary signals under a number of outliers and “heavy tails” disturbances. Addition of wavelons receptor fields, including their transformations (dilation, translation, rotation and form parameters of activation function) allows to improve the network approximation properties, that is confirmed by the experiments research results.

Bibliography

[Bishop, 1995] C. M. Bishop Neural Networks for Pattern Recognition. Oxford: Clarendon Press, 1995.

[Billings, 2005] S. A. Billings, H.-L. Wei A new class of wavelet networks for nonlinear system identification. IEEE Trans. on Neural Networks, 16, 2005, P. 862-874.

[Bodyanskiy, 1987] Ye. Bodyanskiy Adaptive algorithms for nonlinear control plant identification. ASU i pribory avtomatiki, Kharkiv: Vyscha shk., 81, 1987, P. 43-46. (In Russian)

[Bodyanskiy et al, 2001] Ye. Bodyanskiy, V. Kolodyazhniy, A. Stephan An adaptive learning algorithm for a neuro-fuzzy network. Computational Intelligence. Theory and Applications, Ed. by B. Reusch, Berlin-Heidelberg-New York: Springer, 2001, P. 68-75.

[Bodyanskiy, Vynokurova et al, 2006] Ye. Bodyanskiy, N. Lamonova, O. Vynokurova Recurrent learning algorithm for double-wavelet neuron. Proc. XII -th Int. Conf. "Knowledge - Dialogue - Solution", Varna, 2006, P.77-84.

[Bodyanskiy, Vynokurova et al, 2007] Ye. Bodyanskiy, N. Lamonova, O. Vynokurova Double-wavelet neuron based on analytical activation functions. Int. J. Information Theory and Applications., 14, 2007, P. 281-288.

[Bodyanskiy et al, 2007] Ye. Bodyanskiy, I. Pliss, O. Vynokurova A learning algorithm for forecasting adaptive wavelet-neuro-fuzzy network Proc. XIII -th Int. Conf. "Information Research & Applications", Varna, 2007, P.211-218.

[Bodyanskiy, Vynokurova et al, 2008] Ye. Bodyanskiy, O. Vynokurova, V.V. Boyko, S.V. Sushkov, O.O. Pavlov Clinical data analysis in the medical researches based on computational intelligence methods, Kharkiv, Eksklusiv, 2008. (In Russian)

[Chui, 1992] C. K. Chui An Introduction to Wavelets. New York: Academic Press, 1992.

[Cichocki, Lobos, 1992] A. Cichocki, T. Lobos Adaptive analogue network for real time estimation of basic waveforms of voltages and currents IEEE Proc, 139, Part C , 1992, P. 343-350.

[Cichocki, Unbehauen, 1993] A. Cichocki, R. Unbehauen Neural Networks for Optimization and Signal Processing. Stuttgart: Teubner, 1993.

[Cook et al, 1996] D.J. Cook, B.K. Reeve, G.H. Guyatt et al. Stress ulcer prophylaxis in critically ill patients. Resolving discordant meta-analyses. JAMA, 1996, 257, P. 308-314.

[Daubechies, 1992] I. Daubechies Ten Lectures on Wavelets. Philadelphia, PA: SIAM.,1992.

[Holland, Welsh, 1977] P.W. Holland, R.E. Welsh Robust regression using iteratively reweighted least squares. Commun. Statist., Theory and Methods., 46, 1977, P. 813 – 828.

[Hunt et al, 1996] K. J. Hunt, R. Haas, R. M. Smith Extending the functional equivalence of radial basis function networks and fuzzy inference systems. IEEE Trans. on Neural Networks, 7, 1996, P. 776-781.

[Itakura, 1975] F. Itakura Maximum prediction residual principle applied to speech recognition. IEEE Trans. on Acoustics, Speech, and Signal Processing, 23, 1975, P. 67-72.

[Jang, Sun, 1993] J. S. R. Jang, C.T. Sun Functional equivalence between radial basis function networks and fuzzy inference systems. IEEE Trans. on Neural Networks, 4, 1993, P. 156-159.

[Leonard, 1992] J. A. Leonard, M. A. Kramer, L. H. Ungar Using radial basis functions to approximate a function and its error bounds. IEEE Trans. on Neural Networks, 3, 1992, P. 614-627.

[Li, Chen, 2002] S.-T. Li, S.-C. Chen Function approximation using robust wavelet neural networks. Proc. of the 14th IEEE Int. Conf. on Tools with Artificial Intelligence, 2002, P. 483- 488.

[Mitaim, Kosko, 1996] S. Mitaim, B. Kosko What is the best shape for a fuzzy set in function approximation? Proc. 5th IEEE Int. Conf on Fuzzy Systems “Fuzz-96”, V. 2, 1996, P. 1237-1213.

[Mitaim, Kosko 1997] S. Mitaim, B. Kosko Adaptive joint fuzzy sets for function approximation. Proc. Int. Conf. on Neural Networks “ICNN-97”, 1997, P. 537-542.

[Moody, Darken, 1989] J. Moody, C. Darken Fast learning in networks of locally-tuned processing units. Neural Computation, 1, 1989, P. 281-294.

[Moody, Darken, 1988] J. Moody, C. Darken Learning with localized receptive fields. In: Proc. of the 1988 Connectionist Models Summer School, Eds. D. Touretzky, G. Hinton, T. Sejnowski, 1988, San Mateo: Morgan-Kaufmann, P. 133-143.

[Narendra, Parthasarathy, 1990] K.S. Narendra, K. Parthasarathy Identification and control of dynamic systems using neural networks. IEEE Trans. on Neural Networks, 1990, 1, P. 4-26.

[Park, Sandberg, 1991] J. Park, I. W. Sandberg Universal approximation using radial-basis-function networks. Neural Computation, 3, 1991, P. 246-257.

[Poggio, Girosi, 1994] T. Poggio, F. Girosi A Theory of Networks for Approximation and Learning. A. I. Memo № 1140, C.B.I.P. Paper № 31., Massachussetts Institute of Technology, 1994.

[Rey, 1978] W.J.J. Rey Robust Statistical Methods. Lecture Notes in Mathematics, Berlin-Heidelberg-New York: Springer-Verlag, V. 690, 1978.

[Reyneri, 1999] L. M. Reyneri Unification of neural and wavelet networks and fuzzy systems. IEEE Trans. on Neural Networks, 1999, 10, P. 801-814.

[Sunil, Yung, 1994] E. V. T. Sunil, C. Sh. Yung Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. IEEE Trans. on Neural Networks, 5, 1994, P. 594-603.

[Vynokurova, Pavlov et al, 2007] V.V. Boyko, S.V. Sushkov, O.O. Pavlov, O.A. Vynokurova The anesthesia and intensive therapy strategy in the acute gastro ileum hemorrhage, Kharkiv, Eksklusiv, 2007. (In Russian)

[Welsh, 1977] R.E. Welsh Nonlinear statistical data analysis. Proc. Comp. Sci. and Statist., Tenth Ann Symp. Interface. Held at Nat'l Bur. Stds. Gaithersburg, MD, 1977, P. 77-86.

[Zhang, Benveniste, 1992] Q. H. Zhang, A. Benveniste Wavelet networks. IEEE Trans. on Neural Networks, 3, 1992, P. 889-898.

[Zhang, 1997] Q. H. Zhang Using wavelet network in nonparametric estimation. IEEE Trans. on Neural Networks, 8, 1997, P. 227-236.

Authors' Information

Bodyanskiy Yevgeniy - *Doctor of Technical Sciences, Professor of Artificial Intelligence Department and Scientific Head of the Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, Ukraine 61166, Tel +380577021890, e-mail: bodya@kture.kharkov.ua*

Major Fields of Scientific Research: Hybrid Systems of Computational Intelligence

Pavlov Oleksandr - *Candidate of Medical Sciences, Senior Researcher, Doctor-Anesthesiologist, Government Institution "Institute of General and Urgent Surgery (Academy of Medical Sciences of Ukraine)", Balakireva av.,1, Kharkiv, Ukraine, 61018, Tel. +380577021890, e-mail: pavlov73@list.ru*

Major Fields of Scientific Research: Anesthesiology, Computational Intelligence in Medical Problems

Vynokurova Olena *– Candidate of Technical Sciences, Senior Researcher of the Control Systems Research Laboratory, Associate Professor of Information Technologies Security Department, Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, Ukraine, 61166, Tel +380577021890, e-mail: vinokurova@kture.kharkov.ua*

Major Fields of Scientific Research: Hybrid Wavelet-Fuzzy-Neuro-Techniques, Computational Intelligence.

INFORMATION TRANSFORMATION SYSTEM BASED ON MAPPING OF GRAPH STRUCTURES

Margarita Knyazeva, Vadim Timchenko

***Abstract**: The paper develops and illustrates an approach to transformation of information represented as graph structures. It also provides the conceptual information transformation system based on mapping of graph structures. The system comprises three levels which makes it possible to adapt the transformer to information of different domains. Each level has its own models for describing information and specifying ways of its transformation. The transformer can operate with both structural and textual representations of information. The paper describes the general architecture of the transformer and its components. It offers models describing graph structures and mappings which are specifications for graph transformation. It also exemplifies descriptions of graph structures and syntax restrictions which are imposed on the former. The paper provides a fragment of description of one mapping of graph structure on the other. This approach is implemented in the prototype that ensures program translation from one procedural programming language into another. The prototype supports such languages as Pascal, C, languages of structure program models and is developed within the framework of the program transformation system. The prototype is implemented in the Java programming environment.*

***Keywords**: Transformation of information; rule-based transformation of graphs; structure mapping; structure editing.*

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

Introduction

Computer processing of information has proved to be one of the crucial activities in the majority of applied and theoretical domains. It comprises such tasks as acquisition, engineering and usage of various types of data and knowledge.

There is a constant information exchange among computer systems, their components, people and organizations. The types of information include artificial languages (programming languages, specification languages, data structure languages, etc.), knowledge, data and software represented in these languages. During their transmission, some aspects of information can change: representation format (changes in structure, model or language of representation), interpretation, level of detail.

This can be exemplified by information transformation at different stages of development of information systems. Conceptual data models independent from implementation are mapped on their relevant data models that are implementation oriented in a concrete operational environment. For instance, “entity-relationship” semantic model is mapped on a relational data model: UML language representation converts to XML language representation [Jean-Luc Hainaut, 2005].

Highly demanded task of translating programs from one representation language into another can be an additional illustrative example. It occurs in the following domains:

– development of systems meant for analysis, optimization and parallelization of programs, especially those working with several languages of source codes. The analyzed program in the source language is translated into the internal representation that is later used for analysis and transformations [Partsch, H., Steinbrüggen, R., 1983; Shteinberg B.Ya., 2004];

– program reengineering that deals with the problem of program translation from outdated languages into new ones;

– development of domain-specific languages (DSLs). Usually, these are compact programming languages for solving specific tasks of a domain as opposed to the general-purpose languages designed to solve computational tasks in all domains [van Deursen, A. et al, 2000]. They are gaining ground due to the ever-increasing number of modern applied tasks.

Construction of translators is always determined as a hi-tech and time-intensive task which can be solved only by relevant specialists. This hampers mass development of translation tools.

Many tasks dealing with transformation of different types of information need further research. Various research groups have been developing approaches of different efficiency to their solving. Owing to their complexity, these tasks are mostly regarded to be independent and the methods of their solving are developed separately. At the research stage, checking these methods requires designing prototypes of program systems. Since the problem of program compatibility is not taken into account during the prototyping, designers concentrate on the methods of solving and tend to choose a specific representation of the information used. As a result, the derived computer systems are usually incompatible with each other; they are enclosed to their domains being unable to interact with each other to solve the information transformation tasks usually occurring at joints of domains. The maintenance often appears to be completely unprofitable in terms of time limits and labor input. Thus, they cannot be used to solve tasks of other domains and tasks at the domain joints, which lead to new software development.

Another problem of modeled program systems is that they are not usually developed to the level of distributed network applications (functioning in the Intranet/Internet environment). On the contrary, they are left as desktop personal versions and used only where they have been developed. This sharply limits accessibility, scope of practical application and consequently, demand for such tools that in turn decelerates their development and implementation pace.

Graph data structures are natural and visual tools of representation of complicated structures and processes. This makes it possible to use them widely in computer systems to solve different tasks. In many of these tasks, graphs are used to represent data of heterogeneous structures. Such tasks include representation of abstract syntax trees in translators of programming languages, description of structure and processing of object models of documents, processing of complicated data structures in domains in applied tasks solving, etc.

The majority of models that undergo transformations are represented as graph structures. The transformations are described in terms of graphs and operations and based on rules the performance of which results in a new graph on the basis of the specified source graph.

The aim is to develop models, methods and Internet tools of information transformation represented as graph structures.

This paper contains the description of the conceptual scheme and general architecture of information transformation system, the description of transformation model for graph structures.

The paper was written with financial support from FEB RAS (the Far Eastern Branch of the Russian Academy of Sciences) under Program 2 of the RAS Presidium “Intelligent Information Technologies, Mathematic Modeling, System Analysis and Automation", Project 09-I-П2-04 “Development of Control Systems for Knowledge Bases with Mutual Access".

Concept of Information Ttransformation Based on Mapping of Graph Structures

Let us examine conceptual schema of information transformation based on mapping of graph structures (Fig.1).

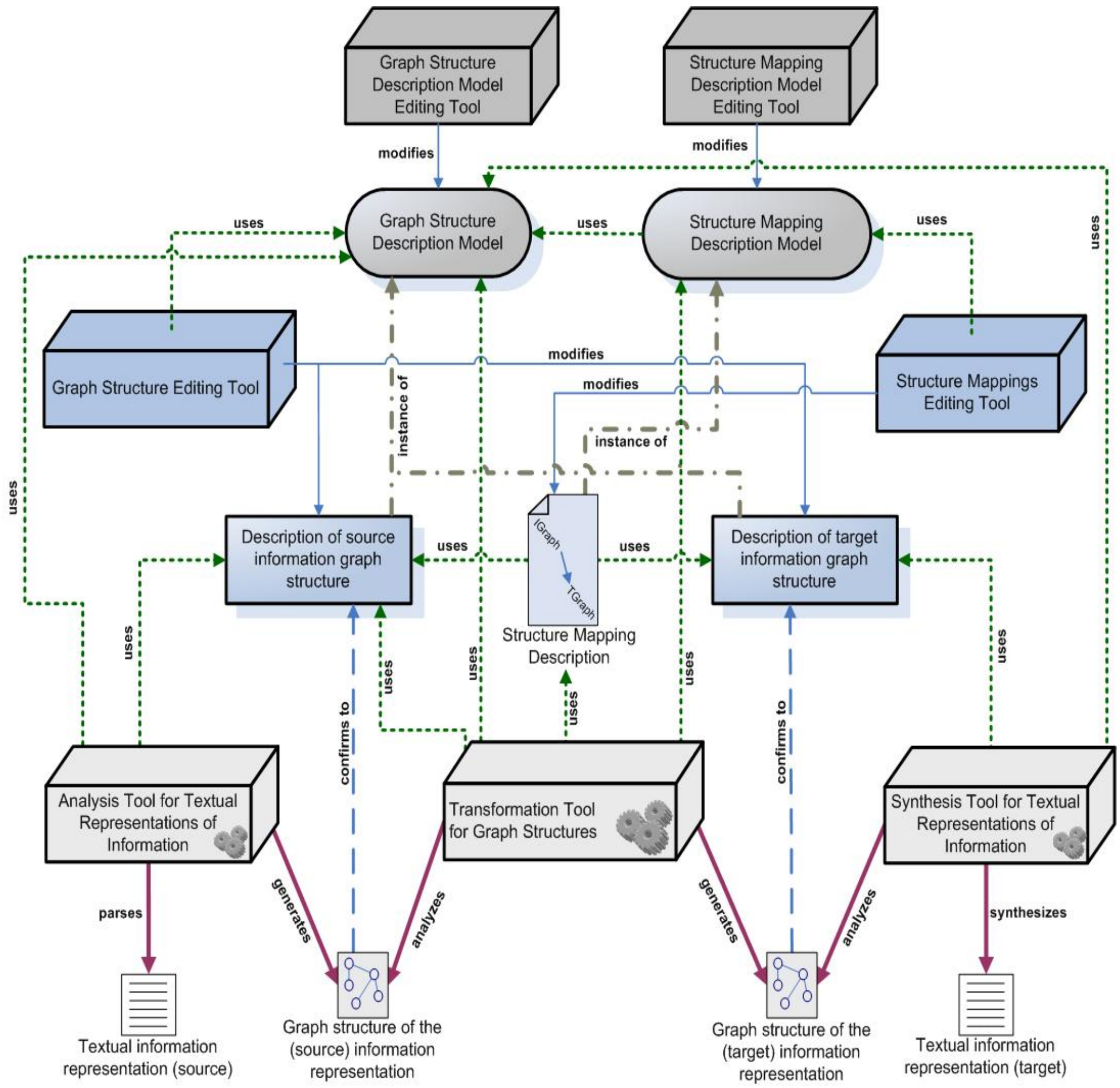


Fig.1. Conceptual schema of information transformation based on mapping of graph structures.

The basic idea of the approach is to represent source information and target information as a graph structure. The correspondent elements of the figure are as follows: *"Description of source information graph structure", "Description of target information graph structure".* This representation comprises:

– definition of a set of concepts of a described domain (graph vertices). Each concept (except those correspondent to the terminal vertices) is defined in terms of a set of some other concepts, and we say that concepts in terms of which some other concept is defined are included in the content of the latter;

– in a set of concepts there is a certain marked concept called an axiom, it does not comprise any arc. The same vertex can include several arcs;

– definition of a set of directed binary relations (with graph arcs corresponding to them) between the concepts. The concepts are linked with each other with the help of directed relations based on the following principle: the concept with an outgoing arc is defined in terms of the concept with an entering arc;

– additional information on a concept or a relation can be obtained from a set of attribute values connected with this concept or relation. A set of related attributes can be empty.

To be able to work with a textual information representation, it is necessary to describe the connection between the graph structure and elements of the syntax particular to the given structure.

This connection determines the content of the correct structures of the language of textual representation from the point of view of its syntax. The connection between the graph structure and its syntax contains such elements of language as punctuation, word order, etc.

This type of information limits the ability of the text to convey the sense given in the graph structure description, therefore this connection will be called syntax restrictions.

Graph structures are described in accordance with *Graph Structure Description Model*. The model regulates, in particular, the rules of specifying syntax restrictions and is used by *Graph Structure Editing Tool* to describe graph structures in terms of the given model. In its turn, *Graph Structure Description Model* is formed and edited with the help of *Graph Structure Description Model Editing Tool.*

Descriptions of the graph structure of information and its syntax restrictions are necessary to carry out syntax analysis of textual representation of information in the specified language and to form this information representation in the form of a graph structure. They are also necessary for the opposite procedure: to synthesize the textual representation of information according to its representation in the form of a graph structure. The procedures are executed respectively by *Analysis Tool for Textual Representations of Information* and *Synthesis Tool for Textual Representations of Information.*

Information Representation as Graph Structure (Target graph – Gt) is the *target* graph structure in regard to *Graph Structure Description* (Metagraph – Gm). This means that the Gt vertices that represent concepts constituting concept contents represented by vertices from Gm are the instances of these vertices from Gm graph structure, whereas the Gm graph structure vertices are prototypes for the Gt graph structure vertices. For example, the Gm graph contains a “name” vertex, while the Gt graph contains “Ivanov” and “Alexandrov” vertices. These latter vertices represent the concepts included into the concept content represented by the “name” vertex from Gt and are the instances of this vertex. Thus, it is possible to assert that Gt is described in terms of Gm.

For example, let there be specified a fragment of the graph structure description representing data on a patient.

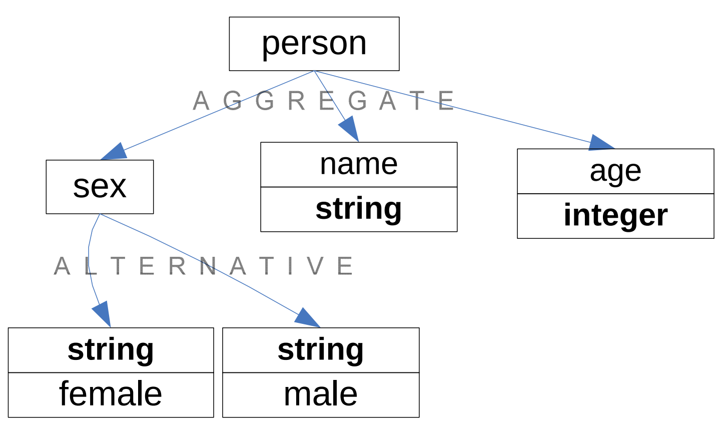


Fig. 2. Fragment of graph structure specifying patient's description.

Description of a particular “Patient 1” patient will be presented as it is shown on Fig.3.

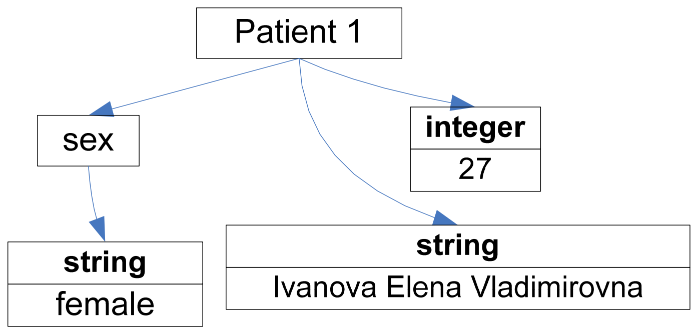


Fig.3. Fragment of graph structure specifying “Patient 1”.

By specifying mapping rules for this information in a particular textual form (graph structure syntax restrictions specifying patient's description) we can, for instance, receive the following representation of “Patient 1”: *Ivanova Elena Vladimirovna, fem., 27.*

The semantics of transformation is specified by *Structure Mapping Description* of the graph structure of the source information representation on the graph structure of the target representation. The mapping the fragments of source and target graph description registers a number of matches between structures. To describe the mappings, there is a language for description of structure mappings.

Structure mappings are described in accordance with *Structure Mapping Description Model*. This model is used by *Structure Mappings Editing Tool* to describe mappings in terms of this model. In its turn, *Structure Mappings Description Model* is formed and edited by *Structure Mapping Description Model Editing Tool.*

Transformation Tool for Graph Structures carries out transformation of information represented by graph structures on the basis of specified *Source Information Graph Description* mapping on *Target Information Graph Description*.

The proposed system served as the basis for the architecture of transformer for graph structures based on structure mapping description.

Let us consider the architecture of transformer for graph structures based on structure mapping description. (Fig.4).

The information component of the architecture comprises the graph structure description model, structure mapping description model, databases for storing graph structure descriptions and structure mapping descriptions respectively.

File system is generally considered to include any read-only memories for file storage available either on a local computer or on a remote one accessible via network interface.

Transformer comprises the following components:

– control subsystem for graph structure transformer;

– structure mapping editor;

– graph structure editor;

– text analysis and synthesis subsystem (TASS);

– graph structure generator.

Control subsystem for graph structure transformer connects components with each other and ensures information transmission and control within these components. It organizes internal interfaces with other system components and loads data from external data sources.

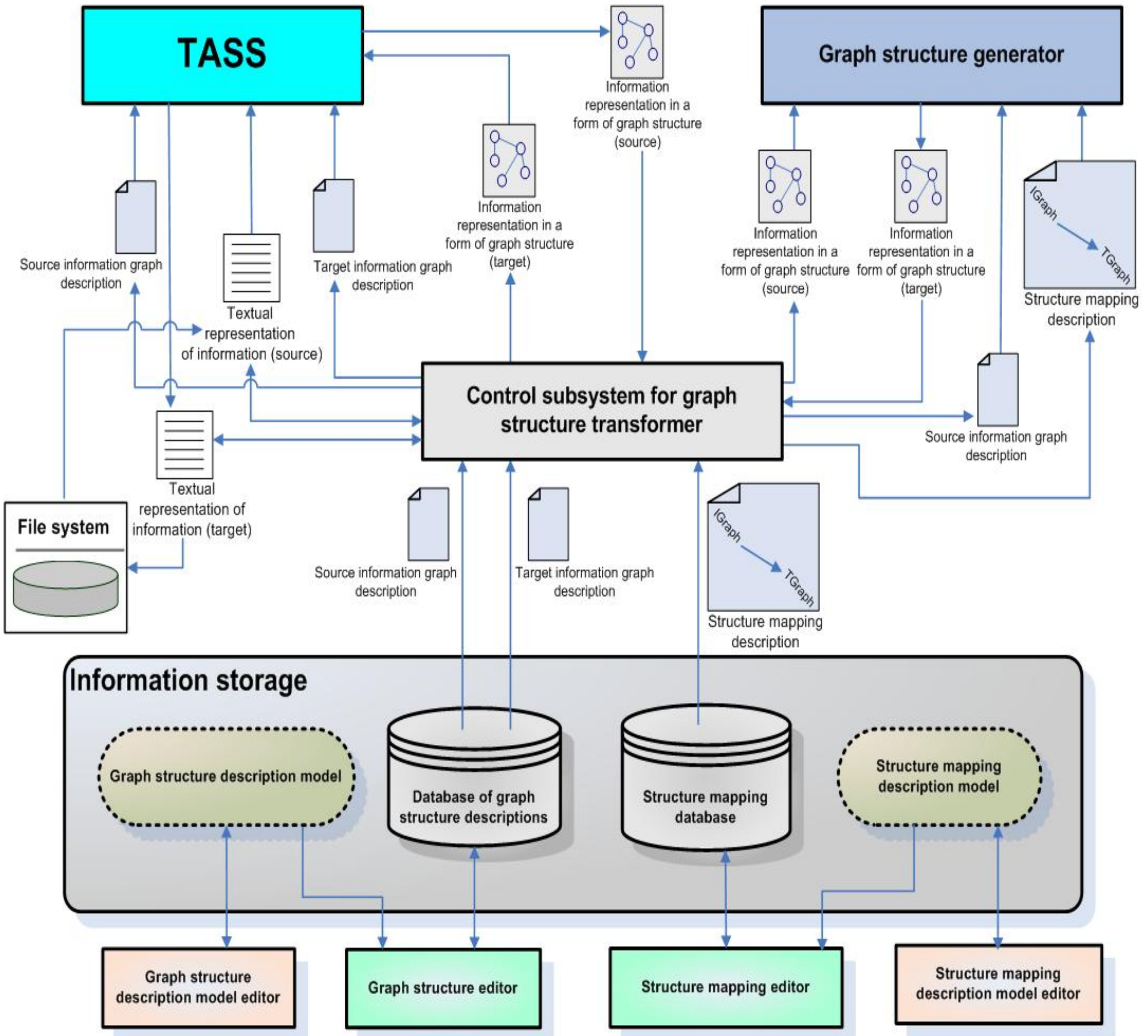


Fig.4. Architecture of transformer for graph structures based on structure mapping description.

Structure mapping editor provides filling and modification of structure mapping database. It is a structure editor the editing process in which is performed in terms of structure mapping description model. This means a certain support of the editing process by the editor itself. The model defines the restrictions on the type and structure of the edited information. This makes the user produce extension of a definition of information and specify values where they are necessary. At each stage, the editor provides the user with all necessary information to proceed to the next editing stage (for example, it offers acceptable variants of information editing where possible).

The structure mapping description model can be modified by structure mapping description model editor. All the modifications will be instantly reflected in the mapping editing (with no reboot of the structure mapping editor required).

Graph structure editor fills and modifies database of graph structure descriptions. This editor (as the structure mapping editor) is a structure editor that executes editing in terms of graph structure description model.

Graph structure description model can be modified with graph structure description model editor. All the modifications will be instantly reflected in the graph editing (with no reboot of the graph structure editor required).

Text analysis and synthesis subsystem is designed to solve the following tasks:

– textual information representation analysis;

– textual information representation synthesis.

Input information for textual information representation analysis consists of:

– description of graph structure of information, according to textual representation of which the graph structure representation must be constructed;

– textual information representation.

Information representation in a form of graph structure is output information.

Input information for textual information representation synthesis consists of:

– description of graph structure of information for which it is necessary to synthesize textual representation according to its representation in a form of graph structure;

– information representation in a form of graph structure.

Textual information representation is output information. As an alternative solution for synthesis task can also be underdefined textual representation, in which not all the words are part of a concrete syntax. Some concepts from graph structure description present in graph information representation can be underdefined. In this case, the target text will contain synthesized names of these underdefined concepts.

Graph structure generator is used to get the graph structure of target information on the basis of the graph structure of source information and structure mapping description.

Input information comprises:

– source information representation in a form of graph structure;

– mapping of source information graph description on target information graph description.

Target information representation in a form of graph structure is output information.

The tool solves the following tasks:

– source and target graph structures navigation;

– search for necessary match: the match for each concept of source graph structure is taken from mapping description; it describes a set of elements (possibly empty) belonging to source information graph structure description and corresponding to the given concept;

– match description interpretation and, as a result of interpretation, in case a set of elements of the target graph structure is not empty, synthesis of the construction described in the match and consisting of these elements.

Graph Structure Transformation Model Based on Mapping Description

Graph structure transformation model based on mapping description *GTM* is a pair (*GDM, MDM*), where *GDM* is a graph structure description model, *MDM* is structured mapping description model.

GDM graph structure description model = (*Concepts, Relations, Attributes, Axiom_Concept, Id_Concept, Const_Concept, Syntax_Restrictions*)

Concepts = $\{Concept_i\}_{i=1}^{conceptscount}$ – finite nonempty set of concepts. Each *Concept* is described by its unique name. Concept name is a nonempty sequence of characters and identifies a certain class of objects of a described domain.

Relations = $\{Relation_i\}_{i=0}^{relationscount}$ – finite, possibly empty set, of relations.

Each $Relation_i$ is a directed binary relation (arc) connecting two concepts which is described as: $Relation_i$ = (*Relation_Name*, *Begin_Concept*, *End_Concept*).

Relation_Name – name of relation that is nonempty sequence of characters.

Begin_Concept – name of concept with a outgoing arc – beginning concept of a relation. *Begin_Concept* $\in$ *Concepts*.

End_Concept – name of concept with an entering arc – ending concept of a relation. *End_Concept* $\in$ *Concepts*.

Attributes = $\{Attribute_i\}_{i=0}^{attributescount}$ – finite, possibly empty, set of attributes. Each $Attribute_i$ is a certain property of a concept and is described as: $Attribute_i$ = (*Attribute_Name*, *Attribute_Argument*, *Attribute_Value*).

Attribute_Name – name of attribute that is nonempty sequence of characters.

Attribute_Argument = (*Attribute_Argument_Type*, *Attribute_Argument_Value*) – attribute argument is described by its type – *Attribute_Argument_Type,* and value – *Attribute_Argument_Value.*

Attribute_Argument_Type = {”Concept”, ”Relation”, ”Identifier”, ”Constant”}.

Attribute_Argument_Value $\in$ *Concepts* $\cup$ *Relations* $\cup$ *Identifiers* $\cup$ *Constants*.

Identifiers – finite, possibly empty, set of all identifiers present in the information representation.

Constants – finite, possibly empty, set of all constants and constant values present in the information representation.

Attribute argument can be a concept (*Attribute_Argument_Type* = “Concept”), a relation (*Attribute_Argument_Type* = ”Relation”), an identifier (*Attribute_Argument_Type* = ”Identifier”) or a constant (*Attribute_Argument_Type* = ”Constant”).

Attribute_Value = (*Attribute_Value_Type*, *Attribute_Value_Value*) – attribute value is described by its type – *Attribute_Value_Type* and value – *Attribute_Value_Value*.

Attribute_Value_Type = {“Concept”, “Identifier”, “Constant”, “String”, “Integer”, “Real”, “Boolean”}.

Attribute_Value_Value $\in$ *Concepts* $\cup$ *Relations* $\cup$ *Identifiers* $\cup$ *Constants* $\cup$ *String* $\cup$ *Integer* $\cup$ *Real* $\cup$ *Boolean*.

String – set of strings.

Integer – set of integers.

Real – set of real numbers.

Boolean – set {True, False}.

Attribute value can be a concept (*Attribute_Value_Type* = “Concept”), a relation (*Attribute_Argument_Type* = “Relation”), an identifier (*Attribute_Value_Type* = “Identifier”), a constant (*Attribute_Value_Type* = “Constant”), a sequence of characters interpreted as a string constant (*Attribute_Value_Type* = “String”), an integer from a set of integers (*Attribute_Value_Type* = “Integer”), a real from a set of real numbers (*Attribute_Value_Type* = “Real”) or by an element of a set {True, False} (*Attribute_Value_Type* = “Boolean”).

Axiom_Concept – a unique concept representing axiom in graph structure information description in terms of which no other concept can be defined, i.e. it has no entering arcs. $Axiom\_Concept \in Concepts$.

Id_Concept – a unique concept representing identifier in graph structure information description that cannot be defined in terms of other concepts, i.e. it is a terminal concept. $Id\_Concept \in Concepts$.

Const_Concept – a unique concept representing constant in graph structure information description that cannot be defined in terms of other concepts, i.e. it is a terminal concept. $Const\_Concept \in Concepts$.

For example, let there be chosen the Milan language description is chosen as information. Graph structure that describes the Milan language can look like it is shown on Fig.5.

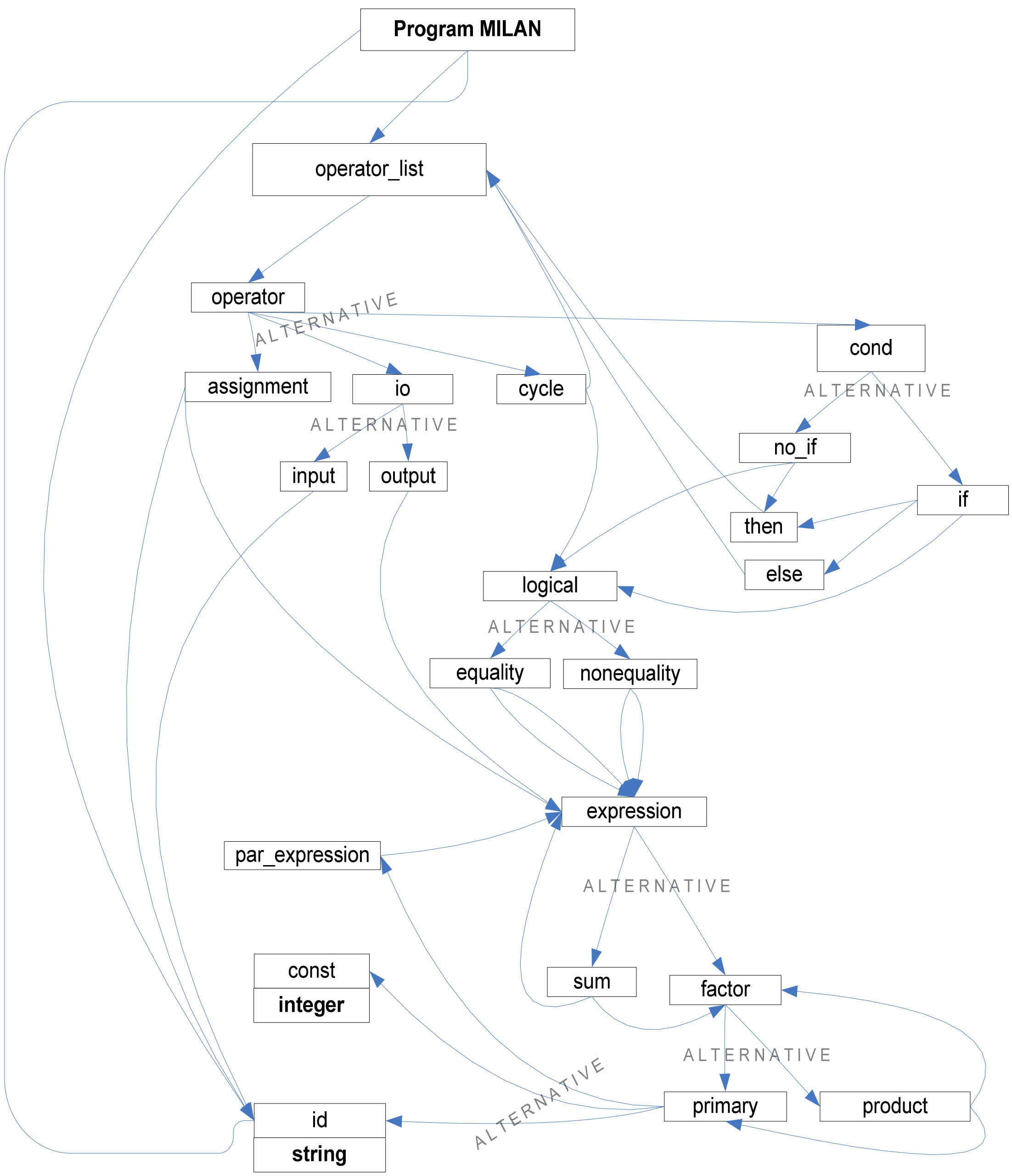


Fig.5. Example of graph structure for the Milan language description.

Program representation in the Milan language in a form of graph structure can look like it is shown on Fig. 6 below. Textual representation of the program is also shown on the figure.

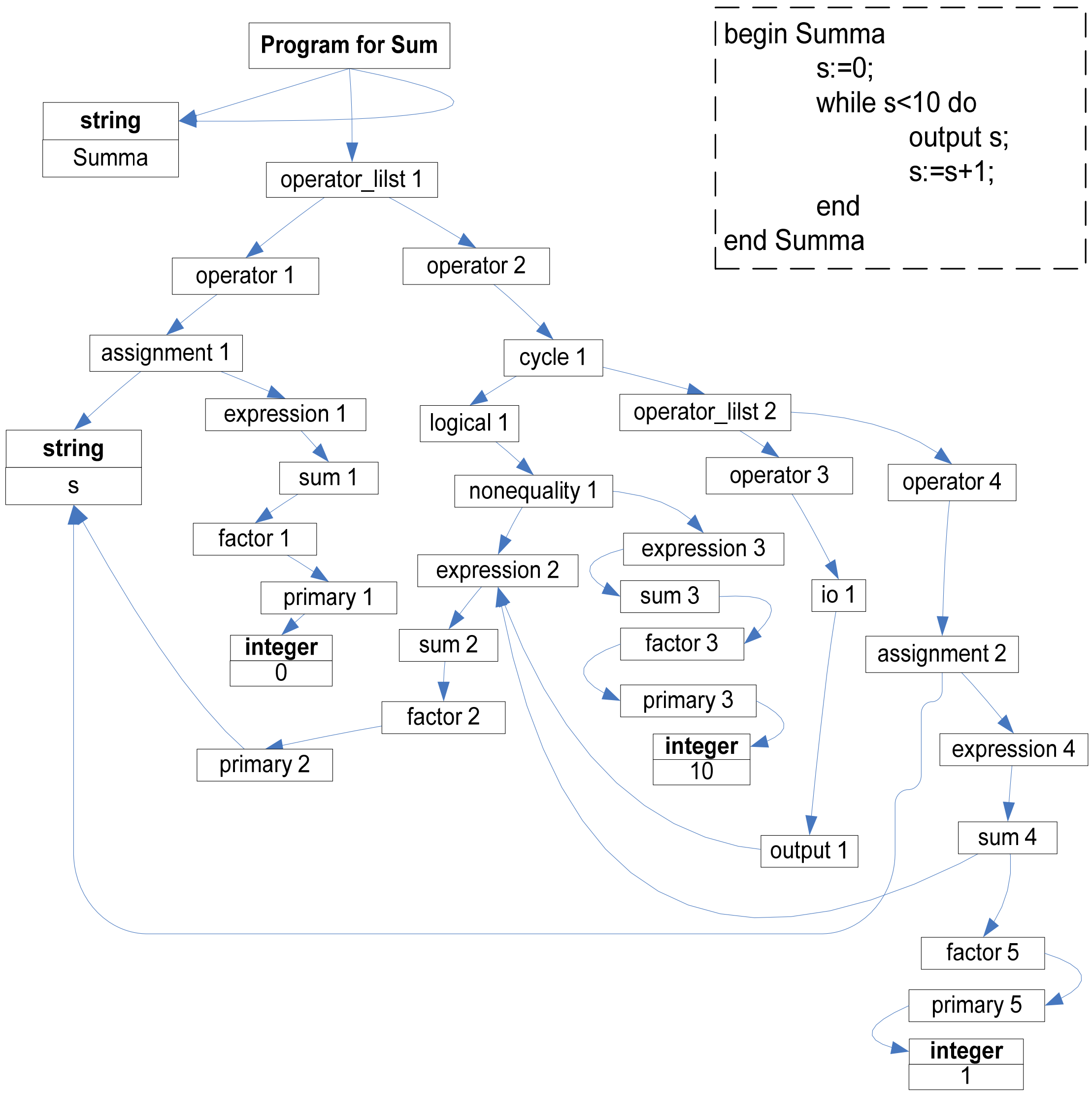


Fig. 6. Example of a program in the Milan language in the form of graph structure.

Syntax_Restrictions = (*Lexica*, *Syntax*) – description of syntax restrictions. It can be absent if it is not necessary to work with textual information representation.

Lexica = $\{<Lexem\_Type_i, Definition_i>\}_{i=1}^{5}$

Syntax = $\{<Concept_i, Definition_i>\}_{i=1}^{conceptscount-2}$

$Lexem\_Type_i \in$ {“Identifier”, “Integer”, “Real”, “String constant”, “String constant limiter”}.

$Concept_i \in Concepts \setminus \{Id\_Concept, Const\_Concept\}$

Definition$_i$ – string of characters that includes metacharacters and represents lexical or syntax restriction for a concrete concept or a type of lexeme.

The syntax restriction model given above has its representation in terms of which the user can formally describe a language of textual information representation. We will further consider the general structure to be used for syntax restrictions representation in any context-free language (CF language). The model of CF language syntax restrictions is shown on Fig.7 below.

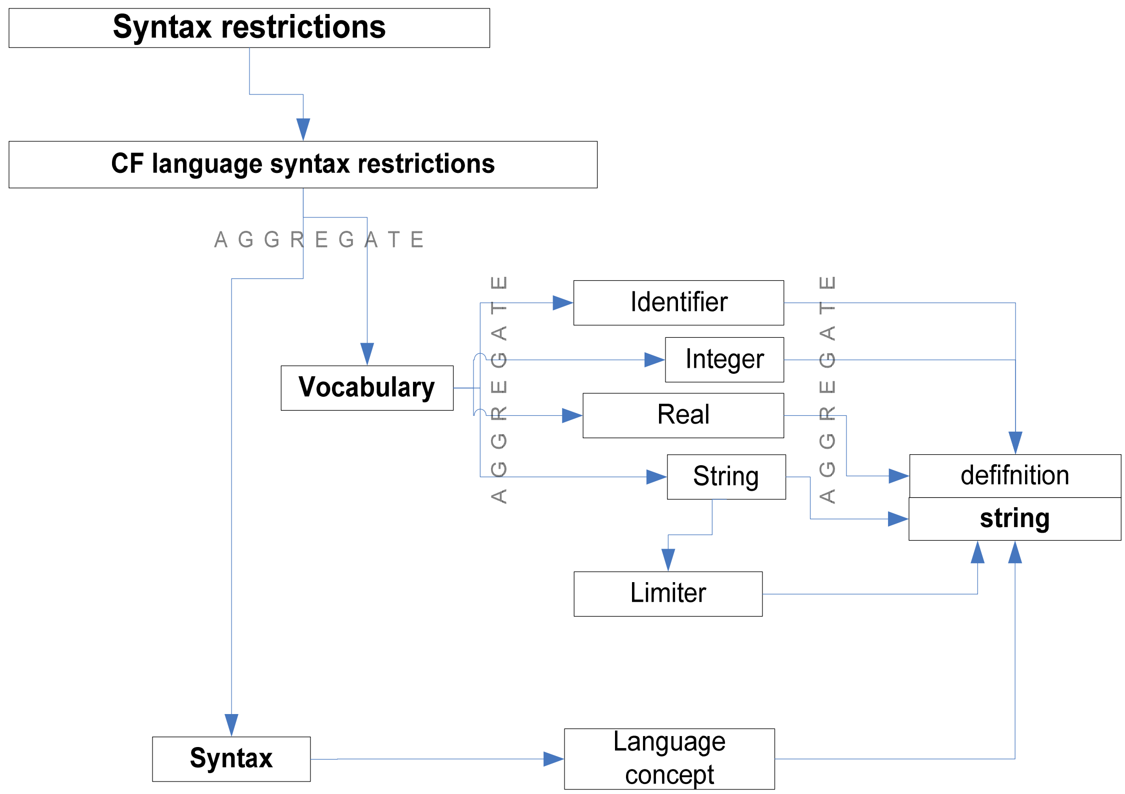


Fig. 7. Syntax restriction model.

Apart from syntax definitions, language syntax restrictions contain lexical definitions as well. For lexis and syntax, there can be a unique restriction specified for each element.

Syntax definition is a definition of a comprehensive concept of graph structure given in terms of comprehensive concepts and elements of a concrete syntax of a textual representation language.

Lexical dictionary of a language is defined by a fixed set of vertices corresponding to base data types – integer, real and string and, apart from it, by identifier type, or name.

Definition terminal vertex describes the *string* kind to which lexical and syntax restriction of a concept represented by a parent vertex corresponds. In concept network that describes syntax restrictions for a concrete textual representation language, the terminal vertex will be a vertex corresponding to the given one. The value of this terminal vertex specifies the concrete lexical and syntax restriction for a concrete language concept or a lexeme type.

All the restrictions on the type of CF language lexemes are specified with the help of regular expressions with the use of perl5 notation.

A string representing a syntax definition contains special metacharacters that carry the following meanings:

“ “ – space is a delimiter character used as a visual aid. Presence or absence of the space do not change the meaning of a writing.

“||” – a metacharacter of concatenation of syntax constructions. Concrete meaning of this metacharacter can be a parameter of synthesis subsystem. For example, it can be settled that syntax constructions in a concrete text would be delimited by the space character.

“?” – question mark. A character of optionality. It means nil or a single entering of an element it follows. This element can be either a concept from graph structure description or a concrete syntax element. Concrete meaning of this metacharacter can be a parameter of synthesis subsystem.

{;} – enumeration metacharacter. Squiggle brackets are used with a language syntax element that can randomly enter the parent element. The first character after squiggle bracket is considered as a delimiter character. Apart from the concatenation character, this character will delimit the iterative elements in a concrete text. If a delimiter character consists of several letters, this character has to be enclosed into double inverted commas. The same refers to a single letter delimiter, for example, {“;”operator} instead of {;operator}. Enumeration restriction for iterative elements is accounted for the fact that only a vertex as opposed to a vertex structure can be in squiggle brackets. The string following the delimiter character up to the second bracket is entirely considered as a name of a certain vertex in the graph structure description.

(…|…|…) – option. It provides the possibility to choose one of many available textual synthesis or analysis variants. Optional variants are delimited by a vertical bar – “|”. All these elements as it was with those of enumeration are considered as vertices' names from a graph structure description. The string containing an option character cannot contain any other construction outside the brackets referring to the specifying option.

«’’» – single inverted commas. They are used to specify elements of semantic network of concepts. Text string enclosed into single inverted commas is considered as a vertex name in the graph structure information description.

«””» – double inverted commas. They are used to specify elements of a concrete language syntax that corresponds to a described syntax restriction.

“\” – backslash. It is used to specify elements of a concrete language syntax that are written is the same letters as service characters. For example, single inverted commas to specify string constants can be written the following way: ”\’”.

Metacharacters do not embrace the lexical restrictions.

Syntax restrictions for graph structure description in Milan language can look like it is shown below on the Fig.8.

Mapping description model (*MDM*) is a generating model that specifies a range of calculations. Every generating model [Uspensky V.A., Semenov A.L., 1987; Kleshchev A.S., 1986] consists of a language formal specification of calculations for this model (language for writing calculation rules) and universal formulation of this model. The latter is defined by the structure of states of generating process, the method of initial state formation, the method of generating process construction based on the calculation rules and initial state, as well as by the generating process stopping rule.

Any calculation specified in the generating model language defines a set of generating processes of graph structures. Thus, a set of generating processes received as a result of universal formulation performance of the generating model for each concrete calculation is considered as a model of every possible generating process of graph structures.

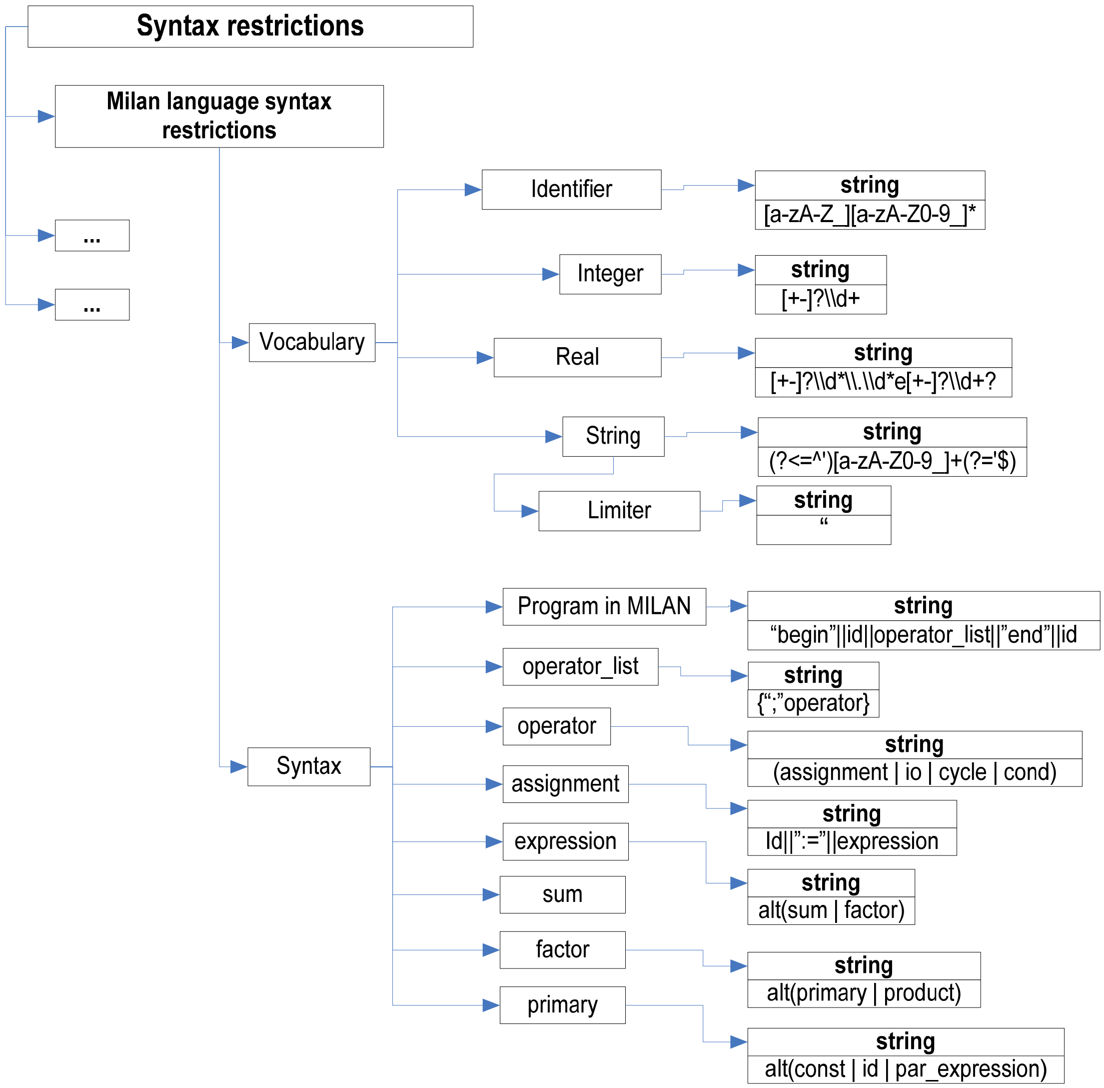


Fig. 8. Example of description of syntax restrictions for the Milan language.

Let us define the language of formal specification of calculation rules for the generating model of graph structures as a group of seven:

(*Concepts*, *Relations*, *Attributes*, *Computable*, *Storable*, *Loadable*, $\{\%\$\}_{i=0}^{n\geq 0}$), where

Concepts – set of concepts,

Relations – set of relations,

Attributes – set of attributes,

Computable = {True, False},

Storable = <*boolean_value*, *alias_name*>,

Loadable = <*boolean_value*, *alias_name*>,

boolean_value $\in$ {True, False},

alias_name – charter string that is a pseudonym for the attribute value under which this value should be saved to the hash-table or by which this value should be received from the hash-table.

Each calculation of the generating model is specified by a finite nonempty set of written in this language generating rules of translation of the source graph structure into the target one: $\pi = \{Rule_i\}_{i=1}^{rulescount}$. Each rule $Rule_i$ is of the form of $\alpha \to \beta$, where $\alpha \in$ *Concepts* $\cup$ *Relations* $\cup$ *Attributes*, $\beta \subseteq$ *Concepts* $\cup$ *Relations* $\cup$ (*Attributes* $\times$ *Computable* $\times$ *Storable* $\times$ *Loadable*). *Concepts* $\cup$ *Relations* $\cup$ *Attributes* $\neq \varnothing$ and for any $\alpha_1, \alpha_2 \in$ *Concepts* $\cup$ *Relations* $\cup$ *Attributes*, if the rules $\alpha_1 \to \beta_1$, $\alpha_2 \to \beta_2$ are included into π, then $\alpha_1 \neq \alpha_2$.

To formulate the rules of the generating model language, there is a structure mapping description language developed. The user can formally describe graph stucture mappings in this language. Each match specifies the composition of concepts, relations and attributes of the target graph structure juxtaposed with the element of the source graph structure.

Specificaiton of abstract syntax of structure mapping description language is given below. The following specification language designations are used [Yershov A.P., Grushetsky V.V., 1977]: **unit** – uniting; [a] – a is not necessary (can be absent); = – concept definition; **:** – position opening, * – possibly indefinite attribute; **ser** – serial component.

Mapping description language = (**ser** mapping description **:** Mapping description)

Description: mapping description language defines a set of mapping descriptions.

Mapping description = (name of graph structure of source information**:** STRING, name of graph structure of target information**:** STRING, matches **:** Matches)

Description: name of graph structure of source information, name of graph structure of target information and matches that describe the connection between them are defined in the description.

Matches = (**ser** match **:** Match description)

Description: This term defines a set of descriptions of matches between the elements of source and target descriptions of graph structures.

Match description = (element of source graph structure**:** STRING, target graph structure construction **:** Target graph structure elements)

Description: In accordance with that, the construction of target graph structure is defined. It has to be juxtaposed to source graph structure element, i.e. a specifically organized set of elements of target graph structure corresponds to a source graph structure element.

Semantics: To interpret the description of the structure of the target graph construction represented in the given match for the specified source graph structure element.

Target graph structure elements = ([concepts **:** Set of concepts], [relations **:** Set of relations], [attributes **:** Set of attributes])

Description: A construction composed of target graph structure elements comprises concepts, relations and attributes specifically connected with each other. Each of these sets can be empty.

Semantics: Make a construction with specified structure of the target graph elements.

Set of concepts = (**ser** concept **:** Concept)

Concept = (name **:** STRING)

Description: *Set of concepts* is a set of target graph structure concepts.

Each made concept is added to the set that contains only those elements that were made by the interpretation of current match description. Before the interpretation of the following match of this set, all of the set elements mush be deleted so that the set becomes empty.

Then each made concept is added to a history of matches between the source graph structure element and a set of target graph structure elements. During the source graph analysis, the history stores the information on made target graph elements, the information on to which source graph element they were juxtaposed. The history provides step-by-step monitoring of the target graph structure generation.

Semantics: To consecutively make concepts a set of names of which is defined by the *Concept* component. For each made concept:

− to add the concept to the set of elements made during the interpretation of current match description;

− to add the source graph element and the concept to the history of matches between the source graph element and a set of elements of the target graph.

Set of relations = (**ser** relation **:** Relation)

Description: *Set of relations* is a set of relations.

Relation = (name **:** STRING, beginning concept of relation **:** Concept, ending concept of relation **:** Concept)

Description: *Relation* is characterized by its name and connects two concepts: one defined by beginning concept of relation selector and the other defined by ending concept of relation selector.

Semantics: If the concept defined by *ending concept of relation* selector has been made by the moment of interpretation of current match description, then

To make a relation with a name defined by *name* selector, specify concept defined by *the beginning concept of relation* selector as the initial concept of the relation, specify the concept defined by *the ending concept of relation* selector as the end concept of the relation.

If the concept defined by *the ending concept of relation* selector has not been made by the moment of interpretation of current match description, then

To add the information on the given relation to the set of not constructed elements. This set stores the history about the elements, structure descriptions of which have been received but they cannot be made because not all of the required structure components of this element have been made.

If during the interpretation of another match the concept defined by *the ending concept of relation* selector appears in a set of elements made as a result of this match interpretation, then this concept is specified as the end concept of this relation after which the relation construction begins. After the relation is ready, this concept is deleted from the set of nonconstructed elements.

Set of attributes = (**ser** attribute **:** Attribute)

Description: *Set of attributes* is a set of attributes.

Attribute = (name **:** STRING, computable **:** LOG, caching value **:** Caching value, value cached **:** Value cached, argument **:** Attribute argument, value **:** Attribute value)

Description: *Attribute* is characterized by its name, argument, on which it is defined, and value it can get. It can be computable and noncomputable. Computable property determines the source of the attribute value: it is either the value of terminal element of the analyzed source graph structure or the value specified during the mapping description. Moreover, attribute value can be saved to the hash-table under a specified pseudonym which is defined by state of the *Caching value* component, and attribute value can be received in the hash-table by entering a specified pseudonym which is defined by the state of the *Value cached* component.

Then, on the basis of the specified information about the attribute value type, the following attributions can be implemented:

If the value type is an identifier or a constant, then attribute value is to be searched among the elements of identifier and constant tables in the source graph structure. If the element with determined value is found, it is assigned to attribute, if it is not, then a constant or an identifier are made, the information on the element is put into a correspondent table and then it is assigned to attribute as its value;

If the value type is a concept or a relation, then attribute value is received as a result of the search among already known (generated) elements of a target graph structure fragment;

If the value type is a string, integer, real or a Boolean value, then the attribute value is assigned during the mapping description, or attribute can get a value of the first found terminal source graph structure element extracted during the search among its elements.

Semantics: To make an attribute with a name defined by *name* selector the argument of which is defined by *argument* selector. To assign *Attribute value* component defined by *value* selector to the attribute as its value.

If the attribute value defined by *computable* selector is *true,* then if the value of *Value cached* component of the attribute defined by *cached* selector is *true,* then take the attribute value from the hash-table by the pseudonym the value of which is defined by *pseudonym* selector.

Otherwise attribute value defined by *value* selector is to be taken from the source graph structure, as the first found value of terminal element of program, therefore at the stage of mapping description it is not necessary to specify this value. If attribute value will be specified during the mapping description and it will be indicated that attribute is computable then the specified value will be disregarded.

Otherwise, if the value defined by *computable* – *false*, then the attribute receives the value specified during the mapping description. If the attribute value is not specified during the mapping description and it is indicated that the attribute is noncomputable, then its value is considered *empty*.

If the value of *Caching value* component of the given attribute defined by *caching* selector is *true*, then save attribute value to the hash-table under the pseudonym the value of which is defined by *pseudonym* selector.

Caching value = (caching**:** LOG, pseudonym **:** STRING)

Description: *Caching value* is a pair the first element of which is boolean *true* or *false* value indicating whether there is a need to save attribute value in the hash-table. The second element is a string of characters that determines pseudonym under which a value must be saved. If the value of the first element is *false,* then the second is disregarded.

Value cached = (cached **:** LOG, pseudonym **:** STRING)

Description: *Value cached* is a pair the first element of which is boolean *true* or *false* value indicating whether there is a need to save attribute value in the hash-table. The second element is a string of characters that determines pseudonym under which a value must be received. If the value of the first element is *false,* then the second is disregarded.

Attribute argument = (type **:** Argument type, value **:** STRING)

Description: *Attribute argument* is a pair (type, value). The type of attribute argument defined by the *type* selector indicates the way of interpreting attribute argument proper, which is defined by *value* selector and determines the acceptable domain for the attribute argument.

Argument type = **unit** (*Concept*, *Relation*, *Identifier*, *Constant*)

Description: *Argument type* can be one of the mentioned.

Argument value = (type **:** Value type, [value **:** STRING])

<u>Description</u>: *Argument Value* is a pair (type, value). The type of attribute value defined by the *type* selector indicates the way of interpreting attribute value, which is defined by *value* selector, and determines the acceptable domain for the attribute.

Value type = **<u>unit</u>** (*String*, *Integer*, *Real*, *Boolean value*, *Concept, Relation*, *Identifier, Constant*)

<u>Description</u>: *Value type* can be one of the mentioned.

Let us fix the source and target information and provide a fragment of a concrete mapping description. On fig. 9, 10 and 11 there is the fragment of the Milan language (source information) mapping description on the Language of structure program models (target information) [Artemieva I.L. et al, 2002; Artemieva I.L. et al, 2003]. The fragment is presented in terms of the mapping description language. On the figures there are matches between “expression”, “cycle”, “sum” concepts of the Milan programming language and their correspondent constructions in the structure program model language.

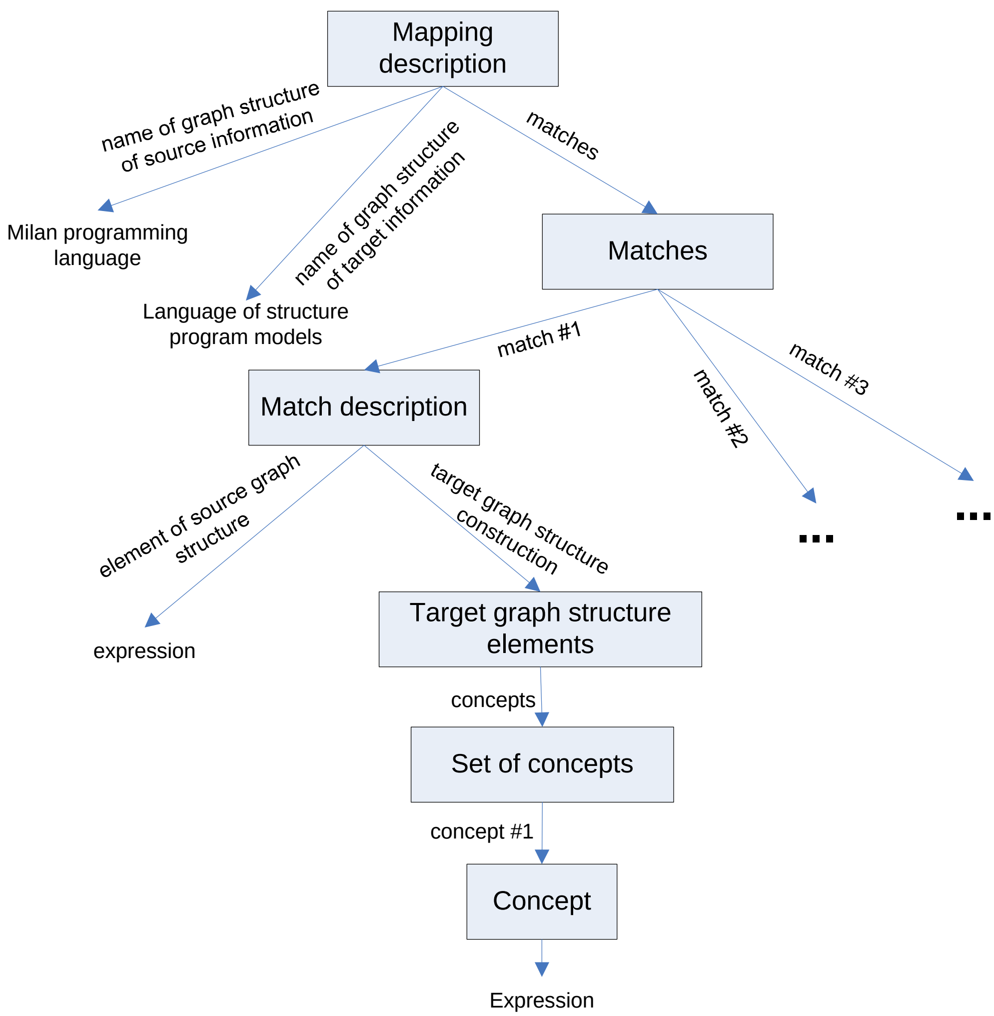


Fig. 9. Fragment of Milan language mapping description on structure program model language. Correspondence description for the *expression* concept

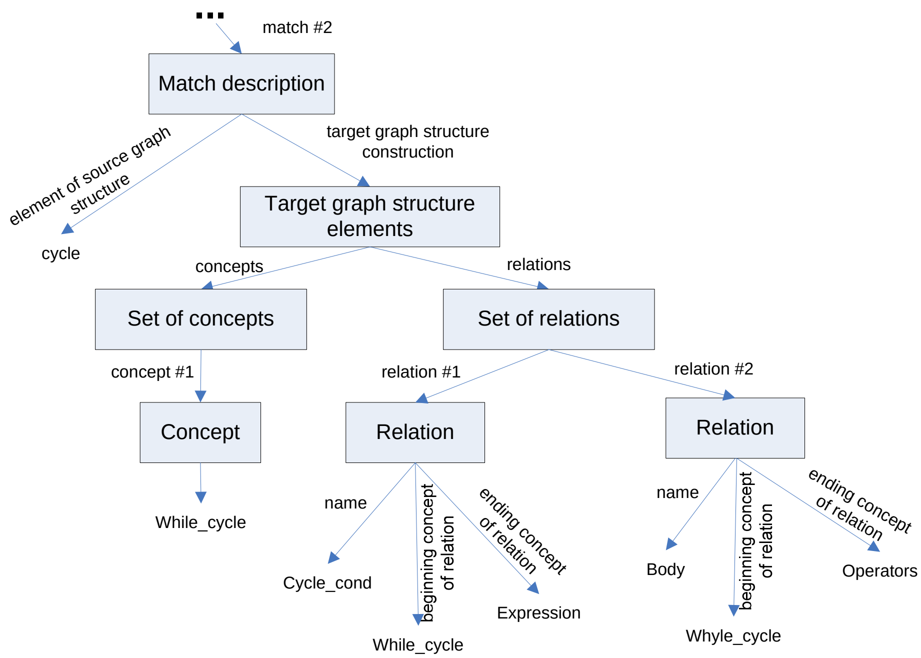


Fig. 10. Fragment of Milan language mapping description on structure program model language (continuation). Correspondence description for the *cycle* concept

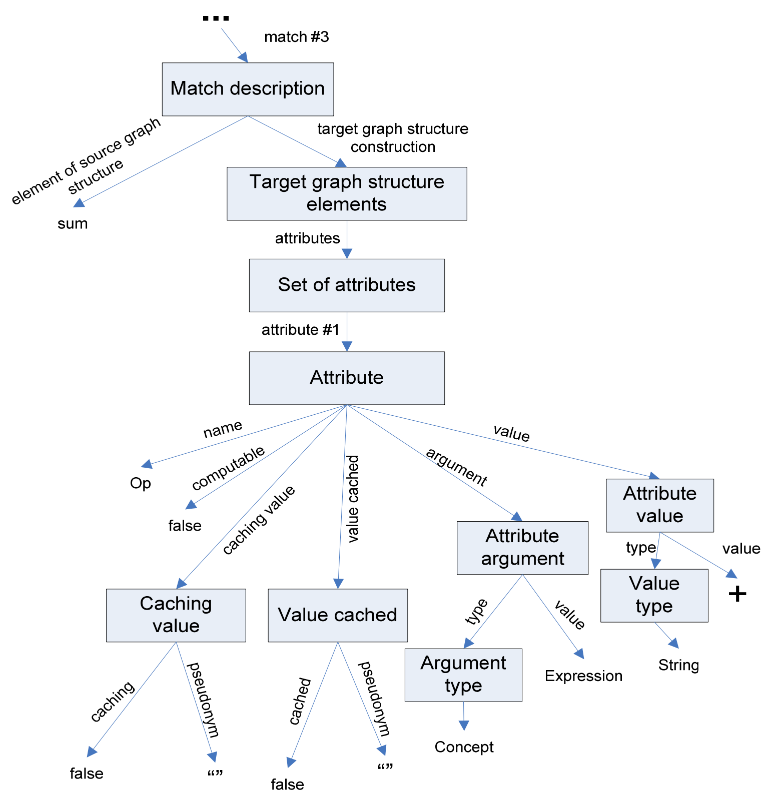


Fig. 11. Fragment of Milan language mapping description on structure program model language (continuation). Correspondence description for the *sum* concept

Conclusion

The paper develops and illustrates an approach to transformation of information represented as graph structures. It also provides the conceptual scheme and general architecture of information transformation system. It offers models describing graph structures and mappings which are specifications for graph transformation.

This approach is implemented in the prototype that ensures program translation from one procedural programming language into another. The prototype supports such languages as Pascal, C, and languages of structure program models developed within the framework of the program transformation system in the specialized bank of knowledge about program transformation. The prototype is implemented in the Java programming environment in the IntelliJ Idea application development environment using resources of the multipurpose knowledge bank (http://mpkbank2.dvo.ru/mpkbank/).

Bibliography

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

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

[Jean-Luc Hainaut, 2005] Jean-Luc Hainaut, Transformation-Based Database Engineering, Transformation of knowledge, information and data: Theory and Applications / Patrick van Bommel, Information Science Publishing (2005), pp. 1–28.

[Kleshchev A.S., 1986] Kleshchev A.S. Semantic generating models. The general point of view on frames and productions in expert systems: Preprint of IACP FESC AS USSR: Vladivostok, 1986. – 39 p.

[Partsch, H., Steinbrüggen, R., 1983] Partsch, H., Steinbrüggen, R. Program transformation systems. Computing Surveys, 15(3), 1983.

[Shteinberg B.Ya., 2004] Shteinberg B.Ya. Open parallelizing system // Mathematical methods of parallelizing of recurrent loops for upercomputers with parallel memory - Rostov-on-Don: Rostov University, 2004.- P. 166-182. (In Russian).

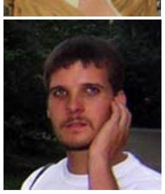
[Uspensky V.A., Semenov A.L., 1987] Uspensky V.A., Semenov A.L. Theory of algorithms: main ideas and applications. – M.: Nauka. Main office of physico-mathematical literature,1987. – (Programmer's library). – 288 p. (In Russian).

[van Deursen, A. et al, 2000] van Deursen, A., P. Klint and J. Visser, Domain-specific languages: an annotated bibliography, SIGPLAN Not. 35 (2000), pp. 26–36.

[Yershov A.P., Grushetsky V.V., 1977] Yershov A.P., Grushetsky V.V. Realization-oriented method of description of algorithmic languages. Preprint, Computational Center of Siberian Branch of Academy of Sciences of the USSR, Novosibirsk, 1977. (In Russian).

Authors' Information

Margarita Knyazeva – senior researcher, Institute of Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences, 5 Radio Street, Vladivostok, 690041, Russia; e-mail: maknyazeva@mail.ru
Major Fields of Scientific Research: Knowledge based problem oriented systems, Software technologies



***Vadim Timchenko** – engineer-programmer, Institute of Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences, 5 Radio Street, Vladivostok, 690041, Russia; e-mail: rakot2k@mail.ru*
Major Fields of Scientific Research: Software technologies, Information systems and data bases

TOWARDS AN INNOVATIVE PRESENTATION AND CREATIVE USAGE OF THE BULGARIAN FOLKLORE WEALTH

Desislava Paneva-Marinova, Radoslav Pavlov, Konstantin Rangochev, Detelin Luchev, Maxim Goynov

***Abstract**: The “Knowledge Technologies for Creation of Digital Presentation and Significant Repositories of Folklore Heritage” research project uses the semantic-based and multimedia digital libraries technologies for innovative presentation, preservation and creative usage of the digitalized Bulgarian folklore knowledge. This paper presents the architecture of the Bulgarian folklore digital library, its main services and components, semantically described folklore objects, types of users, and their activities. The presented structure pursues the powerful and efficient functionalities for effective content structuring, storage, access, search, filtering, maintenance, metadata annotation, indexing, and dissemination.*

***Keywords**: Multimedia Digital Libraries – Functional Specification and Standardization, Multimedia Digital Libraries Architectures, Service-Based Architecture, Bulgarian Folklore*

***ACM Classification Keywords**: H.3.7 Digital Libraries – Collection, Dissemination, System issues.*

Introduction

The tendencies from the past few years in the development of digital libraries with multimedia content lean to transforming the static complex library structures to systems with a dynamic federation of services. This change resulted from the market needs, the emergence of new technologies and from the request for a more strict usage of the existing resources and adapting their content and services to the needs of different user groups. In relation to this, there are many architecture projects and their usage scenarios developed for multimedia digital libraries: service-oriented architecture, grid-based architecture, poor-to-peer architecture, which are presented in more detail in [Pavlov et al., ‘06] [Paneva et al., ‘05].

The “Technologies, Based on Knowledge of the Bulgarian Folklore Heritage” research project (FolkKnow) [Paneva et al., ‘07b] [Rangochev et al., ‘07a] aims to build a digital library of a selection of digitized multimedia objects, taken from fund of the Institute of Folklore of BAS. To achieve this goal, there is a need to research and analyze the variety of architectures, used for implementing the multimedia digital libraries, to evaluate their advantages and disadvantages and to choose the architecture that is most suitable for the specific case. Besides, there is a need for the folklore digital library that has been built for the project to adhere to the European and the world requirements for such an activity and to take into consideration the specifics of the artifacts that it presents.

This article presents the chosen type of architecture for a folklore multimedia digital library, its components, their functional specification, the tools used in the implementation process and the tests made on the system.

Choosing an Architecture for the Multimedia Digital Library of the Bulgarian Folklore

Every type of architecture for multimedia digital libraries [Pavlov&Paneva, ‘05] has a lot of advantages and disadvantages, which have to be considered both in the context of the intended library usage and having in mind the characteristics of the media resources, the necessary semantic and non-semantic descriptions, the resource distribution (in one location or in multiple locations), the desired functionality, the complexity and the time needed to complete the project, the price parameters, etc. Besides, the following factors and characteristics were taken into consideration while choosing the architecture for the multimedia digital library in this project:

- according to the terminology, used in the project, every folklore object represents an aggregation of media objects (one or more), situated in one place;
- the rich internal semantics an the complex relations between the media objects within the folklore objects and between the folklore objects;
- the existence of a wide number of similar objects or objects with similarities in the semantic descriptions;
- the expected variety of the user audience – specialists and non-specialists in the target domain, etc.

Having these factors and features in mind and evaluating the available functionalities from each one of the architecture types, we can conclude that using the grid technologies as a basis for building the folklore digital library would provide optimized access and improved management of the digital resources with the availability of a virtual resource organization, but the complexity of creating a grid-based environment would not justify the results and the used tools. Also, this type of architecture expects to serve distributed computer resources, accessible through local or global networks, which make visualizations through an enormous virtual computer system. This is a condition, which is not needed in the current case, because the media objects are situated in a single location. On the other hand, using peer-to-peer technologies provides a freely-configurable integration of services for digital libraries and sharing of information, dynamic distribution and deployment of complex services and techniques for extracting specific details from multimedia documents, for searching support based on content similarity, etc. In the considered case, all these functionalities are a solution, which does not justify the complex technical implementation, price, effort and software/hardware resources. The limited number of the expected input data for the system on one hand and their rich internal semantics and complex relations between the media objects within the folklore objects on the other hand require the creation of a complex ontological structure for their semantic description. The semantic technologies enable the presentation of the knowledge about the digital objects and collections in the digital library storages via the usage of common classification schemes in the form of ontologies, via the coordination and ordering of the digital collection according to different semantically meaningful criteria, via provisioning of semantic access to them, the semantic-based services, the new ways to distribute resources, the semantic deductions, etc. Besides, in the future we expect the development of new semantics-based approaches, methods and techniques, used in digital libraries as well as improving the existing ones. With the creation of rich semantic descriptions for each folklore object, there emerges the need of improved visualization, accessibility upon request and optimized services for extracting a objects or collections of objects and the existence of a large number of similar objects or objects with similarities in their semantic descriptions requires services for reusing the same metadata and their fast selection.

Considering the prerequisites and the desired functionalities for the project's library and having in mind the specifics of the service-oriented architecture and its purpose to provide the functional modules that serve the main activities for building, provisioning and management of the library content and its semantic and non-semantic descriptions, we can conclude that this type of architecture is the optimum solution in this case. This type of architecture is decentralized, multi-functional, flexible, dynamic and easily-transformable [Rangochev et al., '08]. For the particular case, there is a need to:

- develop the services, which create and manage complex semantic descriptions of the folklore objects;
- provide access to the objects in a more effective and easy-to-use way;
- provide:

  - services for diversity in searching by various semantic descriptors and combinations of theirs as well as optimized result visualization;
  - services for object grouping by various semantic descriptors;
  - services for file format conversion;
  - exporting services for the objects to XML format.
- identify and personalize the users, manage the access permissions, monitor the user actions in the environment, etc.

The conclusions on the subject, made in this section, lead to the formulation of the specific components descriptions of a particular architecture for the folklore multimedia library, oriented towards the provisioning of diverse services.

Semantic Description of the Folklore Knowledge and its Place in the Service-based Architecture of the Bulgarian Folklore Digital Library

Ontology of Bulgarian folklore

Since one of the targets of the FolkKnow project is to present the valuable phenomena of the Bulgarian folklore in suitable virtual form using knowledge technologies, we have to observe and specify the experience that has been gained in the last 500 years in the area of traditional folklore *i.e.* to construct Bulgarian folklore domain ontology [Paneva et al., ‘07a] [Paneva et al., ‘07b] [Luchev et al., ‘08a] [Luchev et al., '08b].

FolkKnow annotator/indexers using this ontology will semantically describe and index the raw audiovisual content in order to create and maintain reusable digital objects. The ontology will be used also to realize semantic-based access to concrete digital objects, representing folklore objects, described by their main features, technical data or context. All this information is included within the Folklore Ontology Concept – the root concept for the ontology.

The process of building of the Bulgarian folklore ontology for the FolkKnow project is necessarily iterative. The first activity is the definition of the scope of the ontology. Scoping has been mainly based on several brainstorming sessions with folklorists and content providers. Having these brainstorming sessions allowed the production of most of the potentially relevant terms. At this stage, the terms alone represented the concept, thus concealing significant ambiguities and differences of opinion.

A clear issue that arose during these sessions was the difficulty in discovering of definite number of concepts and relations between these concepts. The concepts listed during the brainstorming sessions were grouped in areas of work corresponding naturally arising sub-groups. Most of the important concepts and many terms were identified. The main work of building the ontology was then to produce accurate definitions.

Description of the conceptions

The scientific classification and documentation of folklore objects provide folklorists and content generators with a rich knowledge background with plenty of multidimensional data and metadata. There is a special relation among the metadata, which reveals all the knowledge concerning the folklore object obtained from the classification procedure.

The folklore object is related to three levels of knowledge, enriched with a set of sub-levels of the data classification. All these levels of knowledge or "thematic entities" in the ontology conception are supported by the scientific diagnosis results and the related documentation.

The entity "Identification and description" consists of general historical data, identifying aspects such as title, language, archival signature, period, current location of the folklore object, annotation, first level description, second level description, *etc.*

The entity "Technical" includes technical information both revealing the technologies used for folklore object capturing and recording, record situation, record type, record place, record date, main participants in the process (record maker and informant), *etc.*

These main entities and their metadata are supported, documented and provided by the scientific diagnosis, which has been applied to the folklore objects.

Ontological model

We will present the Bulgarian folklore ontological model using classes of concepts, organized in taxonomy and able with properties. Taxonomies are used to organize ontological knowledge using generalization and specialization relationships through which simple and multiple inheritances could be applied. Properties are an interaction between individuals of the domain-classes and the range-classes.

The most representative concepts have been defined first and then they have been specified appropriately in order to get a representation of the knowledge stored in the databases. The Bulgarian folklore ontology is composed of 70 concepts and 82 properties.

Figure 1 depicts a scheme of some relationships between classes of objects in Bulgarian folklore ontology.

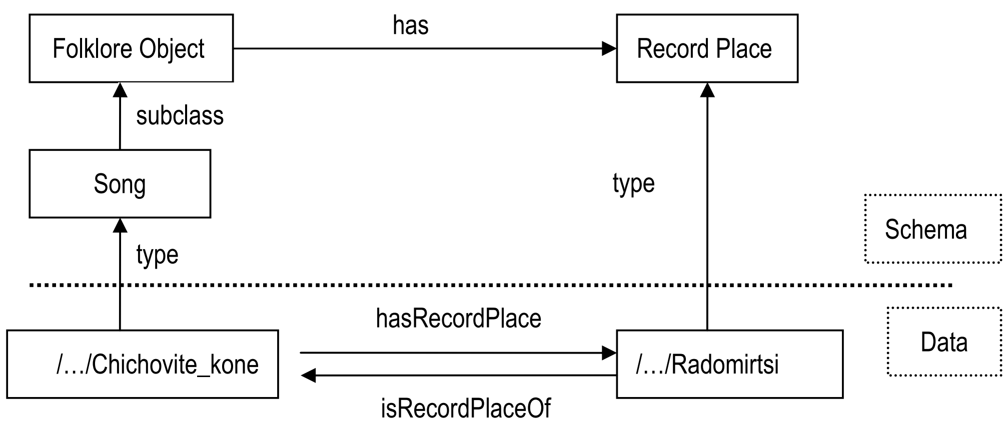


Figure 1: Scheme of relationships between classes of objects in Bulgarian folklore ontology

Main Components of the Bulgarian Folklore Multimedia Digital Library

Basic types of objects and their storage in the architecture of the folklore multimedia digital library

Following are the basic types of objects, which will be managed by the functional modules of the folklore multimedia digital library:

- Media objects (text, images, video, etc.) – they represent digital copies of folklore artifacts from a selected collection of the Institute of Folklore's fund at Bulgarian Academy of Sciences. Along with every media object, there is administrative/technical information about it, which is created in advance during the object digitalization process. The administrative information includes fields like name, media object creator, description, date, type, format, identifier, permissions and source.

- Folklore objects (also known as library objects or digital objects) – the folklore objects represent digital objects, which are annotated and semantically indexed with proper metadata. These metadata show the object semantics according to the "Ontology for the Bulgarian folklore", developed according to module 3 of the project during its first stage. The semantic metadata is used most often in the requests for building information flows in the architecture, which is subject to description.
- User profiles – their purpose is to provide different access levels to different types of library users.

The media objects and the user profiles are kept into a storage of media objects and storage of user profiles, respectively. The metadata, which describe the folklore objects are kept in a storage of semantic metadata (also called data storage). The access and management of the storages, as well as the transition of the different types of objects is provided by a set of functional modules, which implement the library's services.

Main Types of Users of the Bulgarian Folklore Multimedia Digital Library

There are five types of users, defined in the system:

- Administrators – they can create, delete and modify users, as well as review the logs, kept by the tracing module. The administrators do not have access to the library content.
- Editors – this is the group of users with the greatest number of permissions in the library. They can edit all the objects, add and delete objects no matter who entered them.
- Users – the users in this group can view all the objects and can add new ones. They can also edit and delete objects, but only if they have created them.
- Viewers – users in this group can only review the library.
- Guests – the users, which belong to this group can review all the objects

Functional Specification of the Bulgarian Folklore Multimedia Digital Library

Following are the main functional modules of the folklore multimedia digital library [Paneva, '07a] [Rangochev et al., '07a] [Rangochev et al., '08]:

- A module for viewing the content of folklore objects according to their type / rubric, to which they belong;
- A module for creating and editing folklore objects;
- A module for viewing the objects, grouped by different descriptive characteristics (with the option to delete folklore objects);
- A module for searching by:
  - signature and archive number;
  - keywords of the following categories: name, language, annotation, type of the folklore object / rubric;
  - file type;
  - record information (simultaneously or one by one):
    - by situation;
    - by reporter name;
    - by recorder name;

        - by record date;
        - by recording location;
    - Extended search – it provides the option for searching through all the object characteristics;
- A module for managing the user data;
- A module for monitoring the user actions, which keeps track of the following actions:
    - actions, related to working with the system:
        - Registration;
        - Logging in the system;
        - Unsuccessful log-in attempts;
        - Logging out;
        - Changing of the user password;
        - E-mail address change;
        - etc.
    - Actions, related to the object manipulation:
        - Adding an object;
        - Editing an object;
        - Deleting an object;
        - Adding a file;
        - Deleting a file.
    - Actions, related to the content viewing:
        - Review of objects by their characteristics;
        - View of a single object;
        - Searching for objects by characteristics;
    - Other administrative actions:
        - Changing the user's level;
        - Deleting a user;
        - Generation of an XML copy of the data in the system;
- A module for file format conversion;
- A module for generation of XML copies of the objects in the system.

The module for viewing the content of folklore objects according to their type / rubric, to which they belong is available to all the users of the library, except the administrators. The reason is that the administrators of such systems are often people, who don't have any relation to their content. They only do support tasks. The module itself was implemented similar to the Windows OS file browser (Windows Explorer) and KDE (Konqueror), so that it is closer to the familiar user interfaces for viewing hierarchical information. The left side shows a tree of all classes, which inherit "Type of folklore object" and the right side shows a list of objects of the selected class in the tree.

The module for creating and editing folklore objects is used for adding new objects and modifying the information of already created objects. Through it, one can add more multimedia files to an object or delete existing ones.

Searching for information is the most frequent search and therefore the most important operation in a digital library. This is why, there are several different modules for searching by different criteria:

- Searching by a signature or archive number – This search module is useful for finding objects by their archive number (for example, AIF No 200, folder 1, page 57). In general, there is only one search result. In case of incorrect data, it is possible to have several objects as a result.
- Search by a keyword in the object properties – by name, language, annotation and type of the folklore object – Searching is performed simultaneously over all these properties. It is expected that this module is the most frequently used one. This is why special attention has been paid to its optimization.
- Searching by record information – This module is used to find all the objects, which cover some of the following conditions:
  - All the objects, recorded in a given situation, for example an interview, chat/conversation, etc.
  - All the objects, recorded by a given person.
  - All the objects, recorded by a given informer.
  - All the objects, recorded in a given period of time.
  - All the objects, recorded in a given location.
- Searching by file type – This module allows getting a list of all the objects, to which there is a multimedia file attached – audio, video or images. This type of searching uses the database, in which the administrative information is stored instead of the OWL file that contains the ontology.

Most types of searching use SPARQL (SPARQL Protocol and RDF Query Language). This is a language for requests to the RDF and OWL ontologies. The language is in a standardization process by RDF Data Access Working Group as an official recommendation of the World Wide Web Consortium. The SPARQL syntax is similar to the most widespread language for database requests – SQL.

The module for monitoring the user actions is intended to keep logs for the modified and deleted objects by the users, so that in case of deleted data by mistake or wrong entered data, the responsible user can be found. There is also a log for search requests, whose purpose is to enable statistical reports about the search types that are used least and most often. It would allow the removal of the rarely used search types and the priority optimization of the ones that are used most often.

The module for file format conversion was developed to provide the ability to present every file, which is unsuitable for internet preview in a “light” and convenient form for web preview. The module recognizes the “inconvenient” files, it tries to covert them and on success it replaces the original file with the new “lighter” file. On failure, the module keeps the original file in the library.

The module for generating an XML copy of the data is available only to the system administrators. The purpose behind this module is creating a copy, which can be used as an archive copy on one hand and on the other hand it may serve as raw data for other systems, using information from the library.

Implementation of Functional Components of the Architecture of the Bulgarian Folklore Digital Library

A module for adding objects to the FMDL – Adding objects to the FMDL is implemented through filling and sending a form to the web server. Because the great number of fields to fill, the form is not generated thoroughly. Only the necessary fields for the creation of the objects are generated, following the semantic descriptions, presented in the “Ontology of the Bulgarian folklore”, built in the first stage of module 3 of the project. The technology, used for the implementation is AJAX. The user interface passes a request to the server, in which it requires only that part of the form, which according to the user is necessary to create the object. The server processes the request and returns the required fields as a result, which is visualized in the user interface. After all the fields are filled, the user submits the form. The server validates the data and if everything is correct, it adds the object to the data storage. If there is something wrong, it returns a message to the user, relative with the error (usually, an empty field or unacceptable field value). After the server adds the information from the form to the data storage, there follows a check for attached files in the user request. If there are attached files, the server checks if there are file formats, which are unsuitable for web presentation (for example, wav, .doc, .mpg, .avi, .mpeg, etc.) and if it finds such files, the system refers to the module for file format conversion to formats, suitable for web preview. For each of these files, the module for file format conversion tries to convert them. Upon success, it adds the converted file to the library. On failure (which can occur if the added file has some specifics, which the system cannot recognize), it adds the original file to the library. At the end of the object adding procedure, the system refers to the module for monitoring the user actions, where it adds an “object added” event and writes the author (the user, who created the object) and the event date.

A module for editing objects in a FMDL – The module for editing objects works almost in the same way as the module for adding objects. The difference is that the system doesn’t add information about a new object, but replaces the existing information about an object with the new information, provided by the module for editing. Again, the system checks the form for errors, it processes the files (if there are new files added), it changes its data and finally adds an event for modified object through the module for monitoring the user’s activity.

A module for viewing the content of folklore objects according to their type / rubric – This module takes a request from a user, in which the user specifies the property, by which folklore objects must be found. The module refers to the data storage and makes a request for selecting and sorting the objects by this property. The module for monitoring the users actions records the “view objects by” event and adds data about the date, the user and the property, by which objects are listed. The storage processes the request and returns a result, which the system processes and sends to the user. The user interface visualizes the result in a proper manner.

A module for searching – This module allows the user to set a property/properties, by which objects are found.

The algorithm for searching by a single property – The user interface sends a request to the data server specifying the property and its needed value. The module for searching refers to the data storage of semantic metadata with a query for selection and sorting the objects with the needed value of the specified property. The module for monitoring the user actions records the “search” event with the provided search parameters, the date and the user, who performs the search. The storage processes the request and returns a result, which is then processed by the search module and displayed in a proper manner by the user interface.

The algorithm for searching by more than a single property – - The algorithm is analogical to the one, described above with the only difference that the query to the data storage is more complicated – there are multiple selections of objects for each search property and the result is a sorted section of the selection results.

Testing the functional components of the architecture of the folklore multimedia digital library

After an analysis of the means and standards in the technological implementation of the library environment and the functional modules, the following software was chosen:

Operating system: Microsoft Windows Server 2008 x64 Standard

Web server: Apache HTTP Server v 2.2, PHP v 2.2.9

Database management system: MySQL v 5.1 Standard

Tools for the additional modules: FFMPEG, vwWare, HTML, JavaScript, AJAX

Database query language: SPARQL

The functional components of the architecture of the folklore multimedia digital library, described in [Rangochev et al., ‘07a] and [Rangochev et al., ‘08] were implemented and tested for errors and speed on a server platform with the following hardware configuration:

CPU: 2 x Intel QuadCore 2.8 GHz

RAM: 8GB DDR3

HDD: 4 x 500GB, RAID 10 SATA II

LAN: 2 x 1000Mbit

Objects: 220 test objects

Users: 1000 test users

Testing the functional module for adding/editing an object – server response time (average of 50 attempts): 0.0058 s, i.e. in theory, the functional module for adding/editing an object can process about 172 requests per second for each processor core, which makes 172*8=1376 requests.

Testing the module for selecting folklore objects according to their type/rubric – Time for server response: 0.009 seconds per request, i.e. 888 requests per second.

Testing the module for searching by a single property – Time for server response: 0.008 seconds per query, i.e. 1000 requests per second.

Testing the module for searching by several properties – The test was performed with 25 properties (it will happen very rarely). Time for server response: 0.01 seconds per query, i.e. 800 requests per second.

Testing the module for file format conversion – Converting video files: the server sends a response before it converts the video file, because the process is relatively slow. The average time of processing a video file is about 30 seconds, i.e. you can add about 16 video objects per minute. In this way, after adding a video object, its actual recording in the FMDL happens in 30 seconds.

Converting audio files: The server responds before the file is actually processed. The average time for processing an audio file is about 10 seconds, i.e. in theory, a system with such a configuration can process about 48 audio files per minute.

Converting MS Word (.doc) files: the conversion takes place in real time. The average server response time is 0.04 seconds per request, which is about 200 requests per second.

Conclusion

In this paper we presented the main components of the service-based architecture of a multimedia digital library of the Bulgarian folklore. The presented MDL structure is planned to be flexible and directed to the users. The library will be developed and modified to the new requirements and needs, *etc.* after the initial test. The library with its structural components is directed to two basics groups of users. The first group: scientists who make professional researches of Bulgarian folklore: this people need specialized information about the folklore objects which they research – a genre characteristic of the folklore object, methods and location of the record, mode of the record, signature, a file-number, *etc.* ;the second group: users who are not specialists but they are interested in Bulgarian folklore and they want to get specialized information or just want to get to know to classic objects of the Bulgarian folklore. The structure of the digital library of the Bulgarian folklore described in this way gives universal character.

Acknowledgements

This work is supported by National Science Fund of the Bulgarian Ministry of Education and Science under grant No IO-03-03/2006 "Development of Digital Libraries and Information Portal with Virtual Exposition "Bulgarian Folklore Heritage"" from the project "Knowledge Technologies for Creation of Digital Presentation and Significant Repositories of Folklore Heritage".

Bibliography

[Luchev et al., '08a] Luchev D., D. Paneva, K. Rangochev, Use of Knowledge Technologies for Presentation of Bulgarian Folklore Heritage Semantics, International Journal „Information Technologies and Knowledge", 2008, vol. 2, № 4, pp. 307-313.

[Luchev et al., '08b] Luchev D., D. Paneva, K. Rangochev, Approaches for Utilization of the Semantic Web Technologies for Semantic Presentation of the Bulgarian Folklore Heritage. In the Proceedings of the national conference "Bulgarian Museums in the circumstances of the country membership in the European Union". Sliven, 2008.

[Paneva et al., '05] Paneva, D., L. Pavlova-Draganova, L. Draganov, Digital Libraries for Presentation and Preservation of East-Christian Heritage, In the Proceedings of the Second HUBUSKA Open Workshop „Generic Issues of Knowledge Technologies", 14 September, 2005, Budapest, Hungary, pp. 75-83.

[Paneva et al., '07a] Paneva, D., K. Rangochev, D. Luchev, Ontological Model of the Knowledge in Folklore Digital Library, In the Proceedings of the Open Workshop „Knowledge Technologies and Applications", 31 May – 1 June, 2007, Kosice, Slovakia, pp. 47-55.

[Paneva et al., '07b] Paneva, D., K. Rangochev, D. Luchev, Knowledge Technologies for Description of the Semantics of the Bulgarian Folklore Heritage, In the Proceedings of the Fifth International Conference „Information Research and Applications" – i.Tech 2007, July, 2007, Varna, Bulgaria, vol. 1, pp. 19-25.

[Pavlov et al., '06] Pavlov, R., D. Paneva, Toward Ubiquitous Learning Application of Digital Libraries with Multimedia Content, International Journal „Cybernetics and Information Technologies", 2006, vol. 6, № 3, pp. 51-62.

[Pavlov&Paneva, '05] Pavlov, R., D. Paneva, Towards a Creative Exploitation of Digitised Knowledge in eLearning Systems, Paper presented at the Open Workshop „Multimedia Digital Libraries as Content Providers for eLearning Solutions", 10-11 October, 2005, Paris, France.

[Rangochev et al., '07a] Rangochev K., D. Paneva, D. Luchev. Bulgarian Folklore Digital Library, In the Proceedings of the International Conference on Mathematical and Computational Linguistics „30 years Department of Mathematical Linguistics", 6 July, 2007, Sofia, Bulgaria, pp. 119-124.

[Rangochev et al., '08] Rangochev, K., D. Paneva, D. Luchev, Data and Functionality Management in a Folklore Digital Library, In the Proceedings of the International Conference - Slovo: Towards a Digital Library of South Slavic Manuscripts, 21-26 February, 2008, Sofia, "Boian Penev" Publishing Centre, pp. 246 – 250.

Authors' Information

Desislava Paneva-Marinova *– PhD in Informatics, Assistant Professor, Institute of Mathematics and Informatics, BAS, Acad. G. Bonthev Str.., bl. 8, Sofia 1113, Bulgaria; e-mail: dessi@cc.bas.bg*

Major Fields of Scientific Research: Multimedia Digital Libraries, Personalization and Content Adaptivity, eLearning Systems and Standards, Knowledge Technologies and Applications.

Radoslav Pavlov ***–*** *PhD in Mathematics, Associated Professor, Institute of Mathematics and Informatics, BAS, Acad. G. Bonthev Str.., bl. 8, Sofia 1113, Bulgaria; e-mail: radko@cc.bas.bg*

Major Fields of Scientific Research: Multimedia and Language Technologies, Digital Libraries, Information Society Technologies, e-Learning, Theoretical Computer Science, Computational Linguistics, Algorithmic, Artificial Intelligence and Knowledge Technologies.

Konstantin Rangochev *– PhD in Philology, Assistant Professor, Institute of Mathematics and Informatics, BAS, Acad. G. Bonthev Str.., bl. 8, Sofia 1113, Bulgaria; e-mail: krangochev@yahoo.com*

Major Fields of Scientific Research: Ethnology, Folklore studies, Culture Anthropology, Linguistics, Computational Linguistics, Digital Libraries.

Detelin Luchev *– PhD in Ethnology, MA in Informatics, MA in History; Research Fellow at Ethnographic Institute and Museum, BAS; 6A, Moskovska Str., Sofia 1000, Bulgaria; e-mail: luchev_detelin@abv.bg*

Major Fields of Scientific Research: Communities and Identities, Ethno-statistics, Museums and Archives, Digital Libraries.

Maxim Goynov *– Programmer, Institute of Mathematics and Informatics, BAS, Acad. G. Bonthev Str.., bl. 8, Sofia 1113, Bulgaria; e-mail: maxfm@abv.bg*

Major Fields of Scientific Research: Multimedia Digital Libraries and Applications.

METAMODELLING AND MULTILEVEL METADATA AS A BASIS OF TECHNOLOGY THAT IS INTENDED FOR DEVELOPMENT OF ADAPTABLE INFORMATION SYSTEMS

Lyudmila Lyadova

***Abstract**: Methods of creation and development of distributed information systems dynamically adapted to varying requirements of users and service conditions are considered. Described means are based on use of multilevel models. These models represent various parties of system functioning and system domain with various levels of abstraction and from different points of view. System functioning is based on the models interpretation. This approach is realized in the CASE-technology METAS intended for maintenance of all life cycle of dynamically adaptable systems. The basic levels of the system description are the logical level (the system objects description with data domain terms), the physical level (the description of data presentation in a database) and the presentation level (the description of user interface). All models can be changed during system functioning. New models can be developed on the basis of these models. Now the model of reporting, the Web-model and the business-processes model are created. Adaptation possibilities are based on various means of model development such as language toolkit, tools of data re-structuring, means of generation and adjustment of user interface and so on. Software tools of external data import, means of replication of data and models, tools of program components integration and document management subsystem are included in CASE-system too. This technology is based on DSM and DSL: domain specific languages are used as tools for development of domain models. Objects attributes, operations and business rules are described via DSL. The language toolkit is used for DSL creation. It is the base for the multilevel metamodelling. In such a way development of DSL for the specific domain is the first step of information system creation.*

***Keywords**: Adaptable information systems, CASE-technology, metadata, metamodelling, domain-specific modeling, DSM, domain specific languages, DSL.*

***ACM Classification Keywords**: D. Software: D.2 Software Engineering: D.2.2 Design Tools and Techniques – Computer-aided software engineering (CASE); D.2.11 Software Architectures – Domain-specific architectures; D.2.13 Reusable Software – Domain engineering, Reuse models.*

Introduction

The *adaptability* (ability of systems to adapt to changes of environment and functioning conditions) is one of the most important requirements shown to information systems (IS) of different functions. This concept is sufficiently wide: adaptability includes such interrelated nonfunctional requirements as ability to development, flexibility, expansibility, interoperability, etc. [4, 15, 20]. Such wide nature of adaptability makes this concept interesting to research. This property of information systems is also critical in practice of information system development. Adaptability defines efficiency of investments to implementation and introduction of information systems, their operation and maintenance. It guarantees “survivability” of information systems. *Adaptability* is the characteristic defining ability of system to development according to needs of their users and business.

We distinguish among concepts of adaptive and adaptable systems:

– *adaptable systems* are the "easily changeable" systems including means, which would provide their adjustment for new requirements and conditions dynamically (during operation), would facilitate their support;
– *adaptive systems* are systems which change the behaviour automatically according to the changes occurring in their environment ("context"), they are adjusted to changes of environment without application of any means of "manual" adjustment.

Adaptation of information systems is a process of their adjustment for varying functioning conditions and requirements of users and business processes both during creation of new systems, and at support existing. It is possible to consider this iterative process as the major part of information system life cycle.

It is clear that adaptation process needs adequate tools. There are some *general approaches* to development of adaptable information systems. We consider here appropriate technologies of information system development allowing adaptation possibilities during all information system life cycle.

A *software development environment* consists of a platform of tools and infrastructure, a program under development, people that are involved, and a processes those guide and direct all activities. Typically, however, software development environment is used in the sense *integrated development environment* (IDE) which only covers a certain part of the whole process and of the people involved, for example, the programming activity and the programmers. Participation of experts (non-programmer, specialists in Information system domain) in information system development and maintenance is very important for efficiency of life cycle processes. But it is possible only with use of appropriate tools. As software projects become increasingly complex and development processes extend beyond simply the write-compile-execute style of working. The complex interactions of different people in various roles should be managed and their work should be integrated seamlessly into the overall development.

Generally accepted principles of modern technologies of information system development are based on modeling [1, 2, 5, 7, 14]: analysts create models of system domain and its environment and then these models are realized as data and programs by database designers and programmers. All requirements of business and needs of users can be included to these models and must be taken into consideration by development.

Metamodeling and Technologies of Information Systems Development

Model-driven engineering (MDE) is a software development methodology. It focuses on creating *models* (abstractions more close to some particular domain concepts rather than computing (or algorithmic) concepts). This methodology is targeted at increase productivity by maximizing compatibility between systems, simplifying the process of design, and promoting communication between specialists working on the system development. A modeling paradigm for MDE is considered effective if its models make sense from the point of view of the user and can serve as a basis for implementing systems. The models are developed through extensive communication among product managers, designers, and members of the development team.

In MDE the model is built *before* or *parallel* to the Information system development process. Recently the model is created mostly manually, but there are also attempts to create the model automatically: for instance, model can be built as result of documents analysis and text mining (analysis of technical documentation or instructions, job descriptions and etc.). The model can be constructed from the completed system. The data structure and program code are the models of system too. Therefore one important way to create new models is by model transformation. For example, new model can be taken out of the source code; new model can be extracted from data scheme. Model transformations are executed with using languages like DSL (Domain Specific Language).

The first tools to support MDE were the *Computer-Aided Software Engineering* (CASE) *tools* entered the market in 80s. The CASE-tools and various means of augmented Integrated Development Environments have delivered great value to developers and managers now.

The best known MDE initiative is the *Model-Driven Architecture* (*MDA*), which is a registered trademark of Object Management Group (OMG). There are some other projects of development of MDE-oriented tools.

The main idea of *domain-driven design* (*DDD*) is to concentrate attention at the heart of the application, focusing on the complexity that is intrinsic to the business domain itself. The core domain (unique to the business) is distinguished from the supporting sub-domains. Domain-driven design consists of a set of patterns for building enterprise applications from the domain model out. Applying them together will allow building systems that genuinely meet the needs of the business.

The adaptation means can be based on use of the *dynamic programming languages* (DPL), supporting possibility of redefinition of programs and data structure, modernization of programs via extension of their functional components and etc.

Domain-Specific Languages (DSL) are developed for modeling of various subject domains and sub-domains. The idea of using a domain-specific language has gained a lot of interest in the last decade. The use of domain-specific languages has been shown to raise the level of abstraction, increase productivity and ease maintenance and evolution in software development. Using DSL allow to involve non-programmers in information system development at the different stages of Information system implementation process.

Successful development of a modern enterprise system requires the convergence of multiple views and experience: business analysts, domain experts, database experts and developers with different kinds of expertise, interaction designers – all take part in the process of development such a system. The different products with various functionalities must be integrated to create a running system.

Domain-specific multimodeling (DSMM) is a development paradigm where each view is made explicit as a separate domain-specific language (DSL). Every participant of the system development process has a particular language tailored to solve problems specific to its view on the information system. Development of and tooling for a single DSL is well-studied, but the tools of interplay of different languages in a single system are created now. Multiple DSL are required when moving from simple examples to real enterprise applications. Therefore methods and tools support is needed if multiple DSL development is to succeed [8, 12].

The maximum flexibility at information system development is obtained, if system work is under use of models operating its behaviour which can be changed in the system functioning. Models can be organised at various levels of abstraction and platform independence that provides peak efficiency of development process and possibility of models transformation.

Development of the adaptable systems assumes using of the appropriate tools supporting the demands to system properties. Thus, it is possible to consider that property of adaptability is necessary not only for developed systems, but also for means applied for their creation.

The approach to the development of the adaptable information systems based on creation and interpretation of multilevel models and CASE-system based on this approach are presented it this paper.

The model is a “substitute” object of “original” object. The model is in certain conformity with the original system and provides representation about its some properties. The system model is the abstract description in some formal language of system characteristics important from the point of view of the purpose of modeling. The model describes system behaviour too. It is impossible to be limited to creation only one model at system development process. If system is complex and the number of its characteristics is large then describing system by one model will lead to its extreme complexity. The best approach to the development of any not trivial system is to use set of several models which can be almost independent from each other and will allow to make accents on the different parties of system at the decision of various problems of maintenance of its life cycle.

The models can be generally divided into following kinds:

– *static models*, describing structural properties of systems;

– *dynamic models*, representing behavioural properties of systems;

– *functional models*, describing functional properties of systems.

The static model describes system components, their structure, attributes, interrelations between them and operations which they can carry out. The operations of static model are events of dynamic model. They are functions of functional model. The dynamic model describes sequence of performance of operations in the process of system functioning. The functional model describes the transformations which are carried out by system. It opens the maintenance of operations of static model and events of the dynamic.

According to the degree of model abstraction it is possible to divide models into group of

– *conceptual models* representing a higher-level sight of a problem in terms of system domain;

– *models of the specification* defining "appearance" and external behaviour of system;

– *models of realisation* which reflect the internal structure of system, a concrete way of realisation of observable behaviour of system.

Various metamodel definitions exist. We recognise that the models, created by developers of information systems, should be described in any *formal language – modeling language*. We'll consider that the *metamodel is a model of modeling language* applied for formalisation of the system description. The *linguistic metamodel* is a metamodel described with domain-independent language of modeling. The *ontological metamodel* is a metamodel which describes system in domain-dependent modeling language (domain specific language).

The four-level hierarchy of models is a classical variant of metamodeling at information system development (Fig. 1). In this hierarchy each higher level defines language for the description of subordinate level.

Fig. 1. Classical four-level hierarchy of information system models

There are the data describing a state of a subject domain at the level M0, i.e. the *domain state model* of system is described at the level M0.

The level M1 is an *ontological metamodel* for level M0. It contains *domain specific model*. This model represents structure of data, rules and operations, describing the state of system domain. It is represented with *metadata*.

The level M2 defines a *linguistic metamodel* for the levels M1 and M0: at the level M2 there is a *model of modeling language*, used by analysts and developers for the system development. CASE-tools operate with this model too. The forth (the uppermost) level M3 defines *language used for description of metamodels* of level M2.

The number of levels can be changed at development of concrete systems with different tools.

Information systems and development tools can be divided into *some classes* according to the number of levels in created hierarchy of models and to the way of their use at the information system development and roles of models in technologies applicable for the creation and maintenance of information system.

In *traditional information system* (Fig. 2) there is data describing a state of information system domain and conditions of system functioning. Data is contained “*in*” information system (if file system or in database).

This data corresponds to the model of information system domain which can be described in any language, including natural language or graphical notations.

The model of information system is developed by analysts. Then developers (database designers, programmers) realise this model by means of the chosen tools of information system development, programming systems and database management systems.

In case of model change it is necessary to copy and recompile source codes of applications of information system and/or to restructure information system database. More often at the process of information system development and at the information system maintenance developers “leave” from initial model which is not updated at modification of the data structures and algorithms of their processing. “Mismatch” of models leads to various problems.

In *traditional CASE-technologies* (Fig. 3) the domain model is defined formally. It is contained “*in*” CASE-system. This model is described *in terms of a metamodel* which can be defined in any language and realised by means of CASE-system.

The model change conducts to necessity of change of the code generated by CASE-system. As well as in a case of traditional information system, the metamodel is developed by analysts and then it is realised by developers. The change of metamodel involves copying and recompilation of CASE-tools. However such changes occur extremely infrequently.

Modern CASE-systems include tools for creation and editing of models, and they also allows to generate the database (DB) and the main part of information system applications code. The system developed with CASE-tools usually realises all necessary structures of data defined by the system model, provides access to the data in databases and standard user interface for work with them. The program components, realising behavioural and functional aspects of concrete information system, are detailed and refined more often manually.

In case of change of model the CASE-system allows to generate anew code of applications. A program code added by programmers “manually” remains at observance of certain rules of its writing. After “regeneration” of system applications manual completion of a code usually is required.

An advantage of this approach is that time of the working out initial stages is essentially reduced. Besides, conformity between system and model is supported by CASE-tools.

The *information systems operated the metadata* (Fig. 4) provide more powerful possibilities for dynamic adaptation.

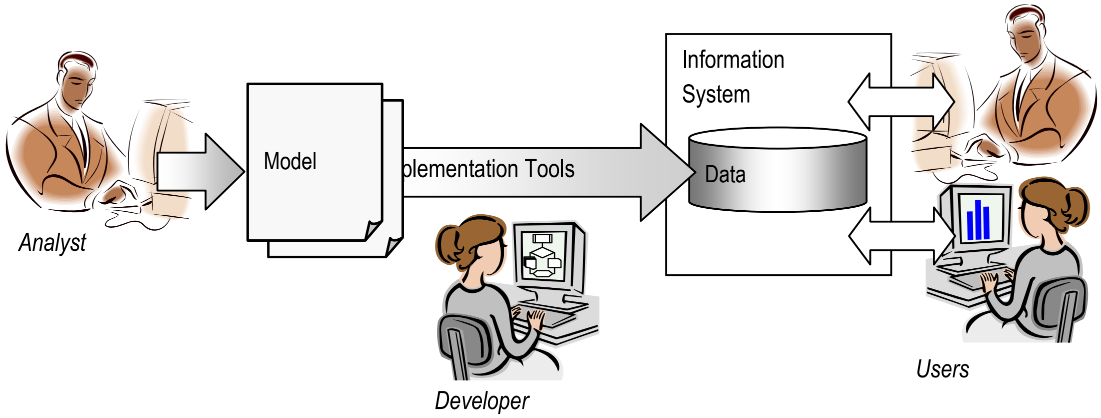


Fig. 2. «Traditional» information system

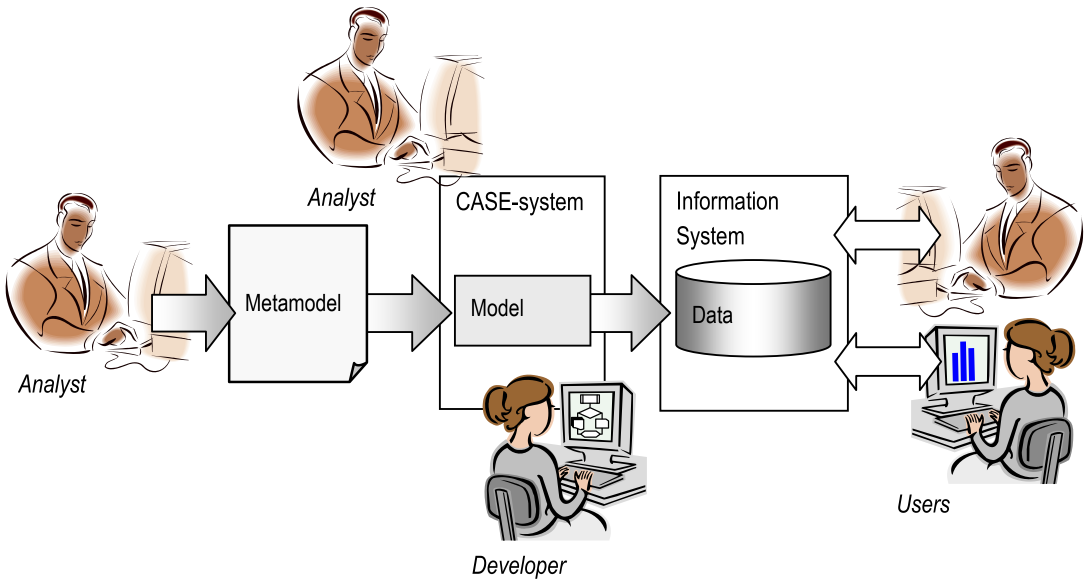


Fig. 3. «Traditional» CASE-system and information system

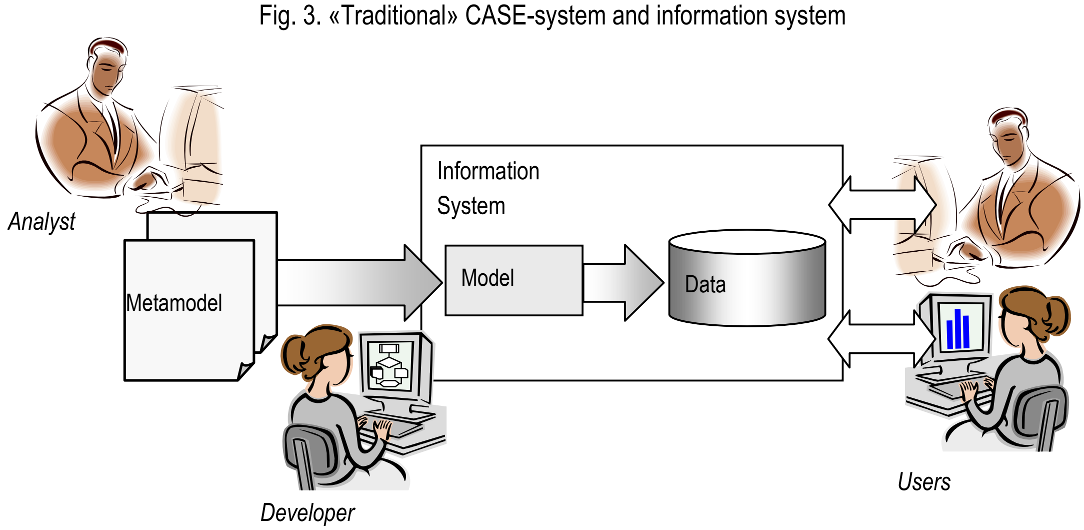


Fig. 4. Information system controlled by metadata

In this case also three levels of the models are used. However the model of a system domain, created by analysts and developers and represented with metadata, is "*in*" information system during its operation and it is used by run-time components of system. Thus, the software of information system acts like a interpreter of model, and model represents a "controlling system", setting rules of information system functioning.

A negative aspect of such approach is that performance of system decreases in its operation. Besides, if there are not possibilities of including to information system of external program components, expanding standard functionality of system, then it is impossible to realise the functions specific for the concrete system, reflecting the particular business processes rules and domain conditions. Therefore the metamodel should be as much as possible powerful and expressive.

The advantage of this technology is that in case of changes of system model it is not necessary to recode and recompile applications of system. It is not required to change source code of programs – the information system simply starts to work according to new model. But "old" data and "new" data can be "miscoordinated" with restructuring. Data loss is possible as result of changes of models.

Technology DSM with code generation provides modeling in terms of information system domain. Models are defined with languages using concepts and relations of information system domain. In case of *domain multimodeling* the specific language of modeling is applied to the decision of each problem in system domain. The domain specific languages are created and used exclusively for each system and problem.

There are two levels of metamodels in this technology (Fig. 5). The meta-metamodel is realised by analysts and developers with Meta-CASE-tools. Metamodel is described on the base of meta-metamodel with Meta-CASE-system. The metamodel defines domain specific modeling language. It is the base for CASE-tools generation. The model of system domain is described with this CASE-system. Then information system (data and applications) is created with CASE-tools on the base of this model. The Meta-CASE- and CASE-tools can be integrated in uniform CASE-system. This CASE-system forms a whole platform of development tools – DSM-platform.

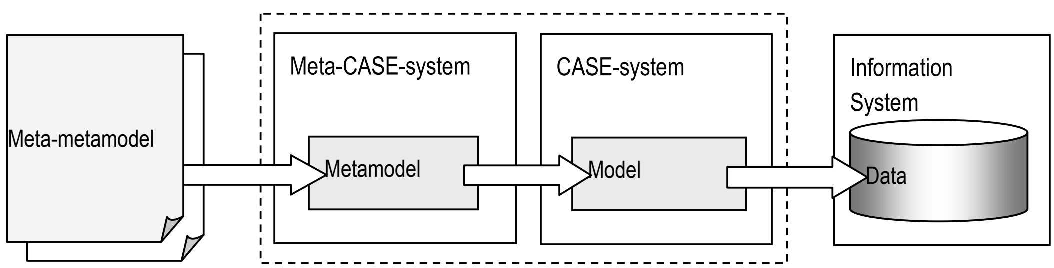


Fig. 5. DSM-technology with code generation

Development of domain-dependent languages (Domain Specific Language, DSL) allows essentially simplifying process of creation of domain models. Experts in the system domain (in applied problems) can take active part in models development process. Other advantages and lacks are related to code generation. They coincide with corresponding characteristics of traditional CASE-technology.

Technology DSM with interpretation of the metadata (Fig. 6) provides the maximum possibilities of dynamic adaptation of information systems. This variant is a combination of two previous. The metamodel, model and data are "*in*" information system developed with CASE-tools based on this technology. In this variant CASE-tools allow to develop models and to interpret them at system operation (special language toolkits and run-time components are included to system for this purpose). In order to use this approach in practice of complex information system development it is necessary that the meta-metamodel was as much as possible expressive and powerful. Interpretation of two levels of metamodels leads at once to conspicuous loss of performance. However at sufficient expressiveness of a meta-metamodel extremely flexible system can be developed.

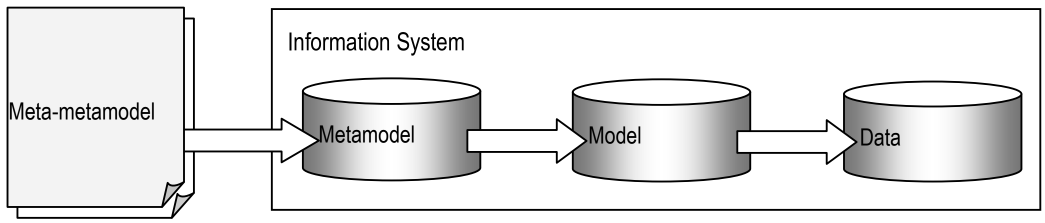


Fig. 6. DSM-technology with metadata interpretation

Such approach is realized in CASE-technology METAS presented in this paper [20]. An experience of some projects demonstrates an applicability of this technology.

The Technology of Development of Dynamically Adaptable Information Systems, Operated under Metadata

The maximum of information systems flexibility can be achieved, if both at system engineering and at its operation the metadata are applied. This metadata describe objects of a system domain, their properties and behaviour, conditions of its work and the characteristics of business processes. The developed information system operates under metadata in interpretation mode. The metadata describe the model of domain state, data and user interface.

CASE-technology METAS (METAdata System) is used for development of dynamically customized information systems operated under the metadata [20]. The base of the adaptability tools is the multilevel model of information system (Fig. 7).

The key aspect of this technology is use of the interrelated metadata describing information system and its environment from the various points of view and at the different levels of abstraction.

In most cases information system development tools work in mode of generation of the system applications in any programming language according to the set of specifications, describing information system domain and environment. The basic difference of METAS CASE-technology from other is in mode of functioning: metadata is used at operating time by program kernel of system in interpretation mode. METAS program kernel (Metadata system kernel, MDK) carries out system functions (data presentation and their processing) according to rules, defined by system metadata, interprets models.

System metadata represents the formalized description of information system. Metadata is stored in special base of the metadata (*MDB*, *Metadata Base*). This metadata, used for system customizing at development and functioning, describe the following aspects of information system: system domain objects and their behaviour, business operations and business processes, primary documents and reports, visual user interface and system security model. The metadata represent *base level models* (*MDK models*), each of which describes a certain part, aspect of information system (some models can describe the same parts of system, but from the various points of view). The metadata in METAS is divided into some interrelated models (Fig. 7). Program components of system work with the metadata of corresponding level (or with the several interconnected levels).

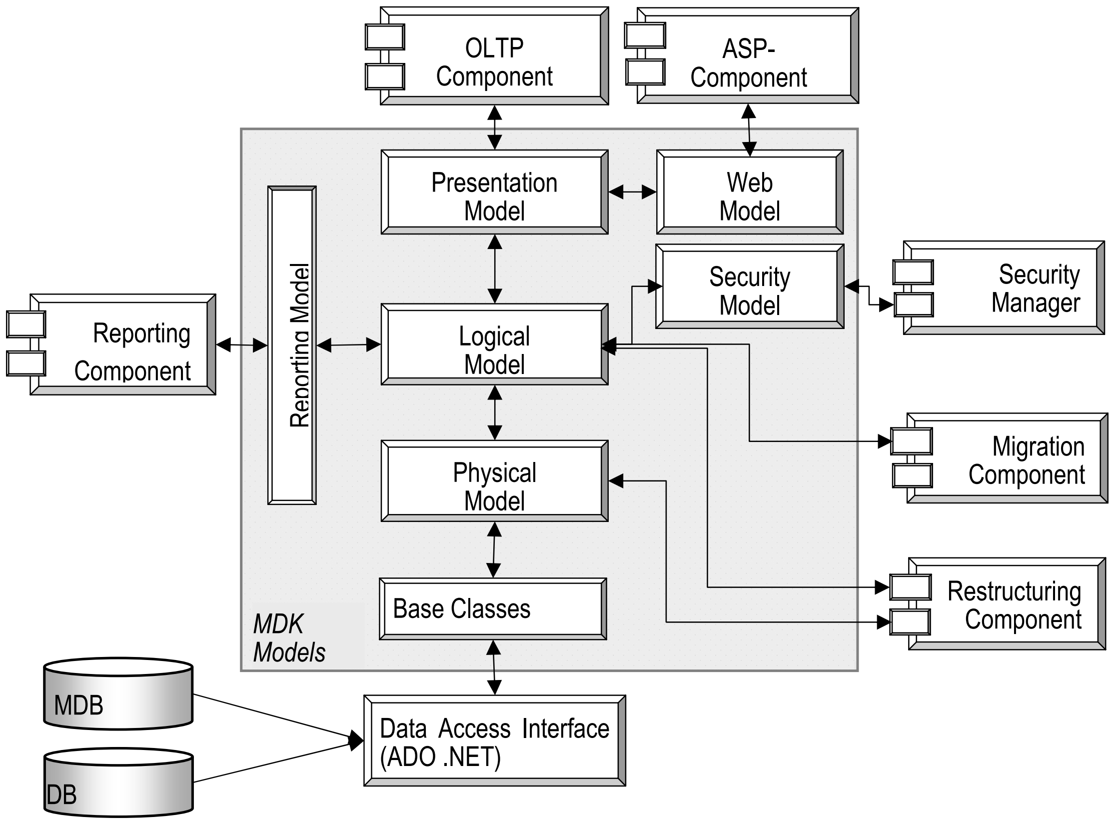


Fig. 7. Metadata models and METAS components

Metadata include three base layers representing following basic MDK models (Fig. 7):

- *Physical model* – the metadata, describing representation of information system objects in a database (DB) of system (for instance, DB scheme (description of tables and interrelations between them, etc.) is stored in this layer). At process of system functioning they form a basis of logical model. This model is automatically generated according to the description of system created in logical level [6, 18].
- *Logical model* – the metadata, describing entities of a system domain, their behaviour (through business operations) and also the general applied operations of information system. This model is based on notations of language UML and allows working in applied domain terms for users of information system. The model of this level is created by analysts and developers. It is the main layer of metadata [6, 10, 18].
- *Presentation model* – the metadata describing the visual user interface at work with objects of information system [10]. This model is automatically generated on the base of logical model. According to this model screen forms are generated. The main form of applications includes tree of objects for easy-to-use navigation and table presentation of basic information on objects of information system. A separate form is generated for all entity of information system. It is used for presentation of detail information on corresponding object and relational objects. Developers and users of information system can customize this model (i.e. user interface) adapt it to their informational needs and preferences.

The set of the characteristics represented in models can be dynamically modified and extended. The set of the models can be extended too via addition of new models describing new parties of information system or new properties of objects or existing, but from the new points of view.

For building algorithms of functioning METAS-system it is necessary to formalize the description with using metadata for all models. Let us consider the formal definition of metadata systems for physical and logical levels in terms of *graph models* [6, 10, 18].

The *nodes* of the *logical model graph* are corresponded to *entities of data domain*. On the physical level all entities are presented with tables of DB (one entity is described by *set of tables*: any entity has one *main table* and *some dictionaries* in addition).

All entities are a *set of attributes*. There are some types of attributes, presented in logical model, listed below:

– *Own attributes*. Own attribute is any one from the *main table* of entity.

– *Dictionary attributes*. Any entity's dictionary has corresponding attribute. When we work with it we may choose necessary values from the list or add a new value in dictionary.

– *External attributes*. If the entity relates to another as "M:1", then it must contain additional attribute with information about this relation.

– *Key attribute*. In fact it is an ordinary own attribute. It contains a link to key field of entity's main table, so it unambiguously determines values of all the rest attributes. This term is introduced only for handy work.

Sometimes it is necessary to represent the main information about real object of some entity in the short form. For example let two entities are related as "1:M". In this case entity on side "M" has an attribute representing another entity (*presentational attribute*). It is not necessary to represent in this attribute all information about object of domain, but only the main, which is enough for its identification. Such main information for entity is determined by the special *presentational expression*. It looks like SQL expression and may contain any attributes of corresponding entity. Any operation and function of SQL language are allowed.

Let us introduce the concept of *relation between entities*. It is analogy with table's relation, but on logical level, in terms of application domain.

There are two basic relation types:

– "1:M". This relation corresponds to physical relation "1:M" between entities' main tables.

– "M:M". This relation corresponds to physical relation "M:M", which is realized using intermediate table in relational database management system.

For any entity relation we may concretely define the count of entity on both sides. They have minimal and maximal number of entity, which may participate in relation. For example the following subtype for "1:M" are possible: "1 : 2..3", «0..1 : 0..1», «0..1 : 1» and other.

The *relations* between entities are presented as *directed arcs* of the graph of logical model.

The main task of logical level is to provide users possibility to work in term of application domain, without thinking about physical information storage in DB. In other words, there is no necessity on logical level to take care about distribution of fields between physical tables. Now user may work in terms of entity, simply calling its name in combination with the name of attribute. Entity builds SQL query itself and either executes it itself (operations of inserting, deleting and updating information in database) or returns ready SQL expression.

The graph presentation of logical model is automatically mapped to graphs of physical and presentational levels.

In addition developers and users can extend logical model with XML.

All new models are based on these models. Creation of new model requires development of programs mapping new model to models of basic levels or interpreting this model.

Now the following models extending basic MDK models are included in system:

– *Reporting model* – the metadata describing queries of users, templates of reports formed at execution of business operations and business processes. This model is used for the analysis of the data and visualizing of analysis results [11].

– *Business-processes model* – the metadata describing business operations and business processes supported by the system.

– *Web-model* provides access to resources of information system for remote users through the Web-interface. This model is the base for development in Web-portal of system.

The *security model* allows to realize monitoring of privileges of users at work with system resources, controlling access rights [3, 9]. The protection subsystem works with own DB. This DB contains information on users and their privileges used for authorized access of users to system data and functions.

The CASE-tools allows to describe objects of information system domain and business processes, to build queries and reports in terms of domain, to customise standard user interface and also to export and import models and the data dynamically. The component of data migration allows to import data from external structural sources with schemes described by users. Users can export all results of data processing to various formats (text, MS Word and Excel documents etc.). *Means of information retrieval* and *documents analysis* allow searching documents and getting information from them. An adaptation of information system is realized without reprogramming of system components and without participation of programmers. The standard business logic can be extended via definition of new types and new operations that are specific for concrete domain.

Language Toolkits and METAS DSM-platform

As it is seemed above use of DSL allows to simplify information system development and to involve experts to process of creation and modification of domain models. Special language toolkits ("meta-CASE-tools") are necessary for development of these languages (meta-metamodels). Meta-CASE- and CASE-tools are integrated for development of systems based on DSMM in order to create multiple interrelational models of information system domain. These models are to be conformed. One approach to models conformation is to use uniform metalanguage for models development. The language used for creation of other languages is named *metalanguage*. Process of models development can be iterative: created languages can be used by developers as metalanguages for creation of other languages and so on. All languages will be conformed on the base of common metalanguage. Such approach is realized in METAS DSM-platform (Fig. 8) that is presented in this article [21, 22, 24, 25].

Now uniform universal DSL does not exist. Different CASE-tools are based on such languages of visual modeling as SDL and MSC – for modeling of telecommunication systems, IDEF, DFD, EPC, BPEL and BPML – for modeling of business processes, etc. Recently for the modeling language standard applies UML.

Key aspect of the DSM-approach is use of DSM-platform including language toolkits, translators or interpreters, etc. The modern DSM-platforms support optimum variants of visual modeling of information system domain and mode of system functioning. The decision on choice of concrete language completely lies on the developers of information system choosing means of system development.

Nowadays some systems for development of DSL exist. They include graphical editors of visual languages and allow possibility of definition of own graphic notations for specialists (experts in different domains and applied problems). Such means are MetaEdit+ [16], MS DSL Tools, Eclipse GMF, State Machine Designer, REAL-IT [19], UFO-toolkit [23] and so on. Technologies Microsoft DSL Tools, GMF, MPS are strongly connected with corresponding DSM-platforms – MS Visual Studio, Eclipse, IntelliJ-IDEA accordingly, therefore the languages created by means of these toolkits cannot be used in other platforms and external applications. These technologies do not allow possibility of creation both visual, and text DSL. Besides, any of technologies listed above (except MetaEdit+) do not allow developing dynamically adjusted languages.

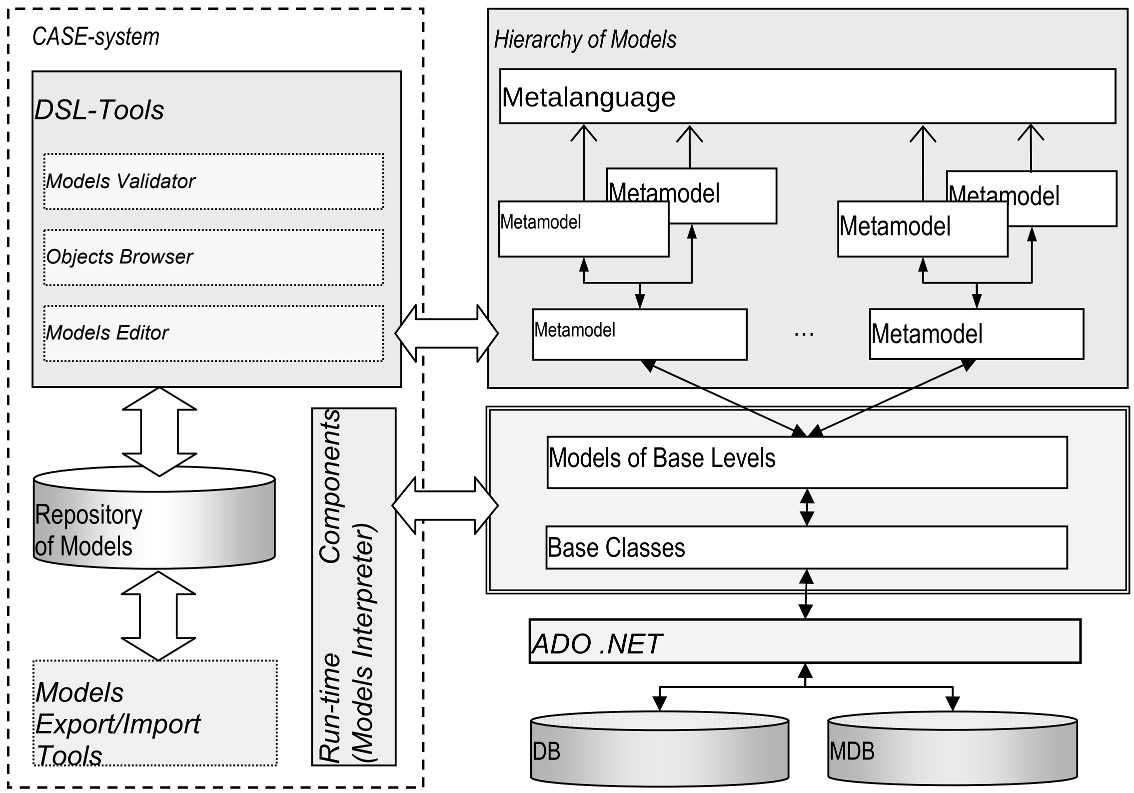


Fig. 8. Simplified structure of information system based on the DSM-platform with (meta)models interpretation

Elimination of noted lacks is the purpose of working out of a metalanguage and DSL-toolkits MetaLanguage – METAS DSM-platform (Fig. 8).

An approach to creation of dynamically changeable languages is based on storage of the language description in the base of metadata (repository) of information system accessible at its functioning.

New possibilities of toolkits and information systems:

- dynamic adjustment of languages for varying conditions of system functioning and requirements of business processes and needs of users;
- work in terms of the information system domain that are habitual for the users;
- reuse created languages and models in similar projects;
- integration of several domain specific languages into one system.

For the language development it is necessary to create its model – a metalanguage. On the base of metalanguage developers can create new languages. They can make changes in existing languages adapting DSL for their needs. If the description of the metalanguage is presented in the form of the metadata a possibility of changes in this description is real too. Besides, process of creation of metamodels becomes iterative and this process can be infinite.

It is necessary to integrate into a developed metalanguage both text, and graphic representation: graphic representation is convenient for using at the description of domain models, and text – at DB creation (for definition of restrictions, for description of templates of reports, etc.).

By analogy to traditional formal languages visual languages can be defined as a set of graphic symbols and formalized rules for development of visual models. Graph grammar represents the traditional approach to the formal description of syntax of visual languages. The most suitable model of internal representation of developed languages at system functioning is multigraph (despite simplicity of its topology). It is possible to describe all operations at development of metamodels in terms of oriented labeled multigraphs.

METAS Language toolkits include graphic editor, objects browser, repository and validator of models. Process of DSL development begins with definition of its metamodel: the basic constructions of language, the relation between them and the restrictions are described. As a result of metamodel creation developer receives new visual modeling language. Developers and users can create new models with this language. It is necessary to check created models with validator before use for system development. Models are stored in the system repository. Browser allows finding suitable models.

The approach described above is realised in the form of library of classes. The primary goal of library consists in maintenance of work with repository systems.

Architecture of Information Systems Developed with METAS Technology

CASE-technology METAS allows to create distributed information systems with client-server architecture of applications (Fig. 9). Distributed information system includes independently functioning subsystems. Subsystems can interact in different modes according to functioning conditions and user's needs. Subsystems are integrated with Simple BizTalk Server – integration component of METAS platform. The integration components support interaction of subsystems applications with some protocols.

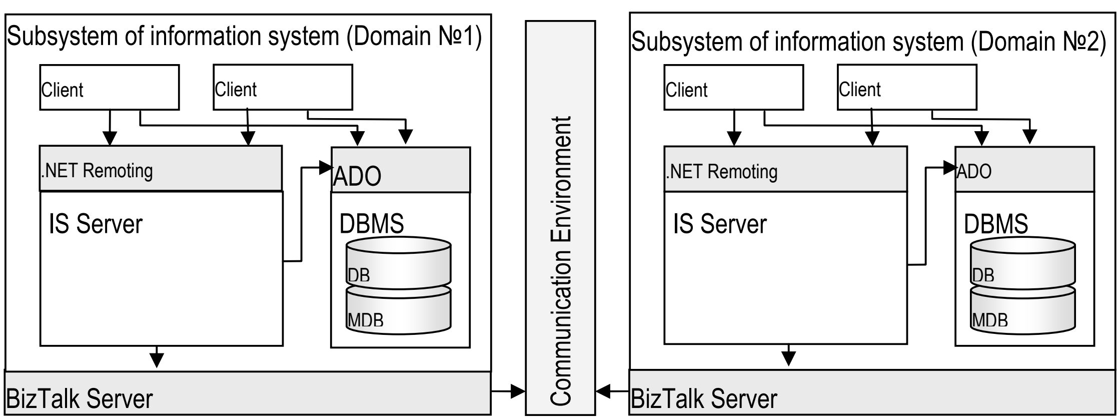


Fig. 9. Domain architecture of information system created with METAS DSM-platform

Simple BizTalk Server allows to replicate system metadata and data. The data replication tools realize two scheme of replication: data and transactions can be replicated.

Documents and templates can be transferred too.

Users of system can set the timetable and choose protocol for the applications interaction.

The components of interaction are the base of means of subsystems conformation and databases synchronization.

Conclusion

The basic advantages of the METAS technology are

– flexibility of systems and possibility of dynamic adaptation to changes of operating conditions, requirements of users with the minimum expenses;

– possibility of integration with external systems via different tools;

– there is no necessity of special training of users: system allows possibility of work in habitual environment in familiar terms of a domain;

– low requirements to a hardware-software platform.

The project of an integrate DSM-platform which provides all stages of information system development since working out of subject-oriented languages of modeling is realized now. Updating of metamodels can be made at any stage of creation DSL and system lifecycle. Thus after modification of a metamodel the system will automatically make all necessary changes to the models created by means of this metamodel. The developed tools form a basis for automatic system adaptation using ontologies in the capacity of domain model.

Acknowledgments

This work was supported by the Russian Foundation for Basic Research (Project 10-01-00794) and by the Russian Humanitarian Scientific Fund (Project 09-02-00373в/И).

Bibliography

[1] Almeida J.P., Pires L.F., van Sinderen M. Costs and Benefits of Multiple Levels of Models in MDA Development // Proceedings of the Second European Workshop on Model Driven Architecture (MDA) with an emphasis on Methodologies and Transformations “Computer Science at Kent”. September 2004. Canterbury, UK. P. 12-21.

[2] Atkinson C., Kühne Th. The Essence of Multilevel Metamodeling // Proceedings of UML 2001 – The Unified Modeling Language. Modeling Languages, Concepts, and Tools: 4th International Conference, V. 2185 of LNCS. Toronto, Canada, October 2001. Springer. P. 19-33.

[3] Chichagova M., Lyadova L. The Application of Graph Model for Automation of the User Interface Construction // International Journal "Information Theories & Applications" Vol.14, Number 2, 2007. P. 189-192.

[4] Chung L., Subramanian N. Adaptable system/Software architectures // Journal of Systems Architecture: the EUROMICRO. Special issue: Adaptable system/Software architectures. Vol. 50, Issue 7 (July 2004). P. 365-366.

[5] Cook St. Domain-Specific Modeling and Model Driven Architecture // MDA Journal, January 2004. P. 2-10. [PDF]. [www.bptrends.com/publicationfiles/01-04%20COL%20Dom%20Spec%20Modeling%20Frankel-Cook.pdf].

[6] Eremina M. About Methods of Mathematical Modelling in the Development of Information Systems // International Journal "Information Technologies and Knowledge". Vol.1, Number 2, 2007. P. 189-195.

[7] Favre J.-M., Nguyen T. Towards a Megamodel to Model Software Evolution Through Transformations // Preproceedings of the Workshop on Software Evolution through Transformations: Model-based vs. Implementation-level Solutions “SETra 2004”. A satellite event of ICGT 2004. October 2004, Rome, Italy. P. 56-70.

[8] Hessellund A. Domain-Specific Multimodeling. Ph.D. Dissertation (preprint), Supervisors: Sestoft P. and Osterbye K. [PDF]. [www.itu.dk/people/hessellund/work/hessellund-thesis-preprint.pdf].

[9] Kourilov D., Lyadova L. Complex Protection System of Metadata-based Distributed Information Systems // International Journal "Information Technologies and Knowledge" Vol.2, Number 2, 2008. P. 116-122.

[10] Kudelko E. The Application of Graph Model for Automation of the User Interface Construction // International Journal "Information Theories & Applications" Vol.14, Number 4, 2007. P. 366-373.

[11] Lanin V. Architecture and Implementation of Reporting Means in Adaptive Dynamically Extended Information Systems // International Journal "Information Technologies and Knowledge" Vol.2, Number 3, 2008. P. 273-277.

[12] Lochmann H., Hessellund A. An Integrated View on Modeling with Multiple Domain-Specific Languages Proceedings of Software Engineering (SE 2009) February 17-19, 2009 Innsbruck, Austria Editor(s): R. Breu. [http://www.docin.com/p-50777268.html].

[13] Mazanek S. Visual Languages. MetaEdit+ : [http://visual-languages.blogspot.com/2007/11/metaedit.html].

[14] Savidis A. Dynamic software assembly for automatic deployment-oriented adaptation // Preproceedings of the Workshop on Software Evolution through Transformations: Model-based vs. Implementation-level Solutions “SETra 2004”. A satellite event of ICGT 2004. October 2004, Rome, Italy. P. 191-198.

[15] Subramanian N., Chung L. Software Architecture Adaptability: An NFR Approach // Proc. Int. Workshop on Principles of Software Evolution (IWPSE`01) / Vienna, Austria. ACM Press, (September 2001). P. 52-61. [PDF] [www.utdallas.edu/~chung/ftp/IWPSE.pdf].

[16] Tolvanen J.-P., Rossi M. MetaEdit+: defining and using domain-specific modeling languages and code generators: [Электронный документ] [http://portal.acm.org/citation.cfm?id=949365].

[17] Tsybin A., Lyadova L. Software Testing and Documenting Automation // International Journal "Information Technologies and Knowledge" Vol.2, Number 3, 2008. P. 267-272.

[18] Борисова Д.А., Лядова Л.Н. Иерархическая модель данных как основа реализации информационной системы, управляемой метаданными // Математика программных систем: Межвуз. сб. научн. тр. / Перм. ун-т. Пермь, 2006. С. 4-13.

[19] Иванов А.Н. Технологическое решение REAL-IT: создание информационных систем на основе визуального моделирования // Сб. «Системное программирование» / Под ред. проф. А.Н. Терехова и Д.Ю. Булычева. – СПб: Изд. СПбГУ, 2004. С. 89-100.

[20] Лядова Л.Н. Технология создания динамически адаптируемых информационных систем // Труды междунар. науч.-тех. конф. «Интеллектуальные системы» (AIS’07). Т. 2. – М.: Физматлит, 2007. С. 350-357.

[21] Лядова Л.Н., Сухов А.О. Метаязык построения визуальных языков моделирования // Технологии Microsoft в теории и практике программирования / Материалы конф. Нижегородский ун-т. Нижний Новгород, 2009. С. 267-273.

[22] Лядова Л.Н., Сухов А.О. Языковой инструментарий системы MetaLanguage // Математика программных систем: Межвуз. сб. научн. тр. / Перм. ун-т. Пермь, 2008. С. 40-51.

[23] Маторин В.С., Маторин С.И., Попов А.С. CASE-инструментарий UFO-toolkit. Автоматизация построения УФО-моделей // Проблемы программирования. 2004. № 2-3. С. 144-149.

[24] Сухов А.О. Предметно-ориентированный язык в адаптируемых информационных системах // Технологии Microsoft в теории и практике программирования / Материалы конф. Новосибирск, 2008. С. 25-26.

[25] Сухов А.О. Среда разработки визуальных предметно-ориентированных языков моделирования // Математика программных систем: Межвуз. сб. научн. тр. / Перм. ун-т. Пермь, 2008. С. 84-94.

Author's Information

Lyudmila Lyadova *– Perm Branch of the State University – Higher School of Economics, Associate professor of the Department of Information Technologies in Business; 38, Studencheskaia St., Perm, Russia; e-mail: LNLyadova@mail.ru.*

THE COMBINED APPROACH TO PRESENTATION OF MULTIMEDIA CONTENT AND TEXTUAL ANNOTATION

Dmytro Nochevnov

Abstract: *The problem of a semantic gap between multimedia content and its textual annotation defined by user is urgent. This paper presents the combined approach to presentation of multimedia semantics which is based on aggregation of several multimedia that are similar by content and textual annotations into the classes. Such classes will contain generalized descriptions of objects and links reflected in the multimedia, and keywords from some thesaurus. To create and update these classes we also describe the hierarchical clustering and machine learning operations. This approach should expand the multimedia search and navigating area due to including the media documents with similar content and text descriptions.*

Keywords*: multimedia data mining, content based multimedia retrieval, text based multimedia retrieval.*

ACM Classification Keywords*: H.3.1 Content Analysis and Indexing - Abstracting methods. H.5.1 Multimedia Information Systems - Evaluation/methodology.*

Introduction

Interest to area of multimedia handling is stipulated by WWW fast growth and arising need of fast media data indexing and search among the big and heterogeneous corpus of documents. Multimedia content producing and using became a usual practice due to existence of effective tools of multimedia indexing and metadata defining. Such metadata can be in machine-readable text format and in the form of visual and audio descriptors (for example in format MPEG-7), and help to handle of multimedia by search engines and intellectual agents [Stamou and Kollias, 2005]. However, the problem of a semantic gap between low level multimedia adjectives, such as color, texture, the scene, etc., and high level concepts like «a mountain landscape», «a misbehavior of the people», which human uses in textual annotation, is still unsolved. Overcoming this gap is one of the Multimedia Data Mining tasks [Stamou and Kollias, 2005], [Petrushin and Khan, 2007]. This issue can be found in traditional multimedia retrieval methods which are usually being divided into two parts [Vihrovsky and Ignatenko, 2006], [Goodrum, 2000]:

1. Search on the basis of textual annotation (Text Based Multimedia Retrieval). Given retrieval methods use high level information as keywords of some thesaurus or textual annotation.
2. Content-Based Multimedia Retrieval which is based on using the low level information such as texture, visual, audio or other kind of patterns in the request.

The main problem of Text Based Retrieval is impracticability of exact and complete multimedia textual annotation which different users can identify in various manners according to their personal experiences and knowledge [Goodrum, 2000]. One of solutions is using the Content-Based Retrieval based on the template of the requested multimedia. However, in some cases such information is insufficient to automatically investigate the information interesting of a seeker, for example the exact time of presented event, the creator of multimedia, objects «behind a frame», three-dimensional relations between the presented objects etc. Also, there is the retrieval precision decreasing which is caused by homonymy, when search results consist of multimedia with different content and similar textual annotations. In turn, retrieval recall decreasing is caused by synonymy which is inversed to homonymy.

Therefore, in our opinion, it is reasonable to combine multimedia low level descriptors and its high level textual annotation for indexing and retrieval [Goodrum, 2000]. It should make more effective multimedia handling and retrieval due to including the information about other multimedia with similar content and textual annotations.

1. Combined Model of Multimedia Knowledge Presentation

It is known that most of the media data contain visual and sound tracks of physical objects written by sensors [Stamou and Kollias, 2005]. If there were several objects of this kind, the information about their space relations (in case of images and video) or time relations (in case of audio and video data) was saved. To mark these objects we can use the existing ontologies or thesauri of objects' names (for example Getty's Art and Architecture Thesaurus [AAT], which consists more than 120000 terms of the art, architecture and other cultural objects, or Library of Congress Thesaurus of Graphic Materials [LCTGM]) to select appropriate terms and save them together with typical objects' metadata. The similar approach to multimedia presentation can be found in [Petridis et al., 2004] where Semantic Web ontology was used to storage of MPEG-7 descriptors. A first order logic can be used to formalize the links between objects.

All of this will allow to automate detection of objects and links between them during indexing of multimedia, and also to expand a search area by engaging of faithful objects.

1.1. Multimedia Content Model

One of approaches to the multimedia presenting is using a semantic net [Petridis et al., 2004], [Dance et al., 1996]. Following it, we will use a directed graph to model objects and links that are reflected in multimedia **m**, and its textual annotation. Such model should consist of:

1) vertex set $v_i \in V_m$, which presents the reflected in multimedia **m** objects; each vertex is associated with some keyword d(v_i) from the thesaurus; named by this keyword objects should be most similar to the represented object by MPEG-7 descriptors' values;

2) edge set $a_{jk} \in A_m$, which corresponds to real links between the reflected in multimedia **m** objects; each edge is associated with predicate $p(a_{jk})$;

3) keyword set T_m, which is defined manually or automatically by indexing of the multimedia textual annotation t_m.

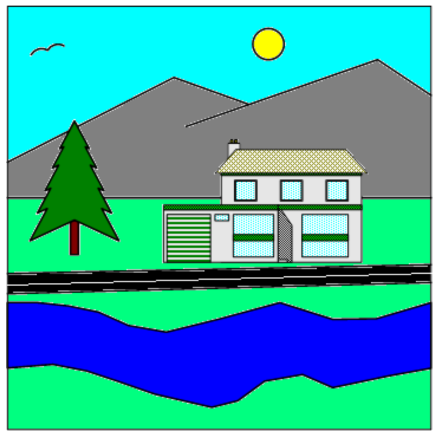
Each keyword d can be associated with one or more vertices of one or more multimedia content models. The same type of link designated by the predicate p, also can be presented in one or more multimedia content models.

Example 1. Let's create a model of image «*Landscape with house in Carpathian Mts*» (see fig.1a). According to the previous definition this multimedia content model **m_1** may consist of the next sets (see fig.1b):

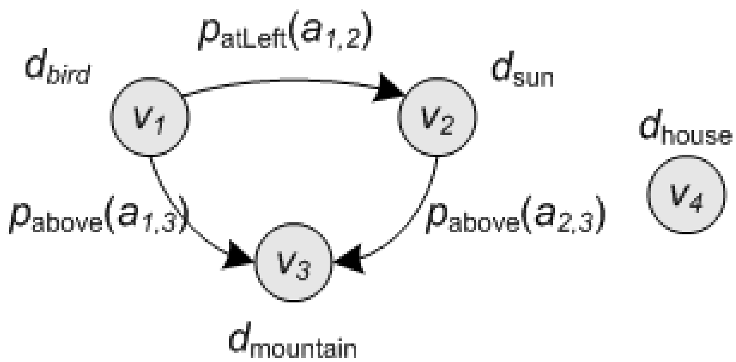
V_m = {$v_1 \Rightarrow$ MPEG-7("bird"), $v_2 \Rightarrow$ MPEG-7("sun"), $v_3 \Rightarrow$ MPEG-7("mountain"), $v_4 \Rightarrow$ MPEG-7("house")};

$A_m = \{a_{1,2} \Rightarrow p_{\text{atLeft}}(a_{1,2}),\ a_{1,3} \Rightarrow p_{\text{above}}(a_{1,3}),\ a_{2,3} \Rightarrow p_{\text{above}}(a_{2,3})\}$;

T_m = {"landscape", "house", "Carpathian Mts"}.



a) image «Landscape with house in Carpathian Mts»



b) fragment of multimedia content model

Figure 1. Example of multimedia content model **m_1**

1.2. Multimedia Class Model

For expanding a search area we propose to aggregate the semantically similar multimedia $m_n \in M_c$ into classes ***c***; such class will include the generalized information about typical objects and links reflected in multimedia, and keywords from their text annotations.

Let's define a multimedia class as directed graph containing:

1) weighted vertex set $v_i \in V_c$, which describe objects reflected in united multimedia $m_n \in M_c$; each vertex is associated with some keyword $d(v_i)$ from the thesaurus of objects' names, and with weight $w(v_i)$; this weight designates the number of objects in set of united multimedia M_c;

2) weighted edge set $a_{jk} \in A_c$, which describe links between objects reflected in multimedia $m_n \in M_c$; each edge corresponds to some predicate $p(a_{jk})$, and to weight $w(a_{jk})$; this weight designates the number of links in set M_c;

3) textual annotation of a multimedia class T_c, defined by indexing of textual annotations t_m of multimedia $m_n \in M_c$; set T_c consists of weighted keywords; each weight $w(t_i)$ designates the number of keywords within the t_m.

Example 2. Let's show an example of multimedia class model c_1 with single multimedia m_1 from previous example 1 (see fig. 2):

V_c = { {$v_1 \Rightarrow$ MPEG-7("bird"), $w(v_1) = 1$},

{$v_2 \Rightarrow$ MPEG-7("sun"), $w(v_2) = 1$},

{$v_3 \Rightarrow$ MPEG-7("mountain"), $w(v_3) = 1$},

{$v_4 \Rightarrow$ MPEG-7("house"), $w(v_4) = 1$} };

A_c = { {$a_{1,2} \Rightarrow p_{\text{atLeft}}(a_{1,2})$, $w(a_{1,2}) = 1$},

{$a_{1,3} \Rightarrow p_{\text{above}}(a_{1,3})$, $w(a_{1,3}) = 1$},

{$a_{2,3} \Rightarrow p_{\text{above}}(a_{2,3})$, $w(a_{2,3}) = 1$} };

T_c = { {t_1 = "landscape", $w(t_1) = 1$},

{t_2 = "house", $w(t_2) = 1$},

{t_3 = "Carpathian Mts", $w(t_3) = 1$} }.

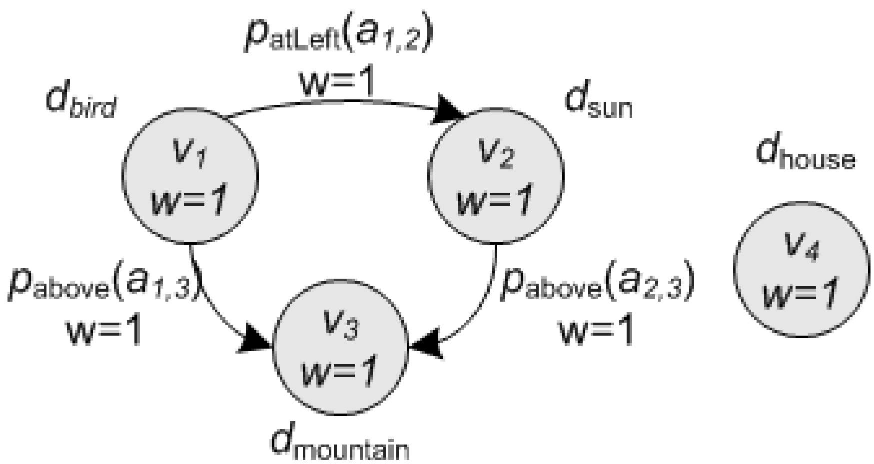


Figure 2. Example of multimedia class model c_1.

2. Multimedia Handling Sequence

We suggest the next operations of multimedia processing:

1. Indexing of a new multimedia and making the multimedia content model *m* in four steps:

a) automatic recognition of objects reflected in multimedia and search the closest by MPEG-7 descriptors values object and take their keywords *d* from the thesaurus of objects;

b) automatic recognition of objects' links and search the appropriate predicates *p*;

c) parse textual annotation t_m to define T_m;

d) aggregate d_m, p_m, T_m into the multimedia content model.

2. Selecting the semantically closest class c_i and updating this one by data about new multimedia; otherwise adding a new class consisting of only new the multimedia.

3. Checking the quality of multimedia class partition and, if necessary, reclustering the semantically homogeneous classes, or all multimedia in corpus.

4. Taking information about the closest by content and annotation multimedia from the selected classes to improve the multimedia retrieval.

The semantically closest class c_i or set of classes ***C*** can be selected manually by user through navigation in the multimedia knowledge base, or automatically by analyzing the distances between multimedia *m* and existing classes $h(m, c)$ by next formula:

$$c = c_k \text{, if } h(m, c_k) = \min(h(m, c_k)) \le h_{\max}, i = \overline{1, N_c} \text{ ,}$$

where N_c - the number of multimedia classes;

$h(m, c)$ – the distance between multimedia and class; will be defined in (1) below.

3. Distances between Multimedia Classes and Instances

Let's define a method of distance between classes and instances of multimedia evaluation. Taking in mind the structure of multimedia content and class models offered above, we will use the similarity measure based on vector of keywords [Ozkarahan, 1986].

Similarity and distance functions can be described according to the [Duran et al., 1974]:

Definition 1. *Similarity function* is nonnegative real function $z(x, y)$ which satisfies the following properties:

1) $0 \le z(x, y) < 1$ for $x \ne y$,

2) $z(x, x) = 1$,

3) $z(x, y) = z(y, x)$.

Definition 2. *Distance function* is nonnegative real function $h(y, x) = 1 - z(y, x)$.

To evaluate semantically close multimedia class it is necessary to define the distance between a class and an instance of multimedia $h(m, c)$. In this case we will take into account the concordance of objects in models, links between them, keywords in textual annotations and their weights:

$$h(m,c) = 1 - z(m,c) = 1 - \frac{\lambda_v \cdot z_v(m,c) + \lambda_a \cdot z_a(m,c) + \lambda_t \cdot z_t(m,c)}{\lambda_v + \lambda_a + \lambda_t}, \quad (1)$$

where $\lambda_v, \lambda_a, \lambda_t \in [0,1]$ - the coefficients of impact, which are defined heuristically;

$$z_v(m,c) = \left(1 - \frac{\sum_{\forall v \in (V_m \cap V_c)} n(w_c(v))}{card(V_m \cap V_c)}\right) \cdot \frac{card(V_m \cap V_c)}{card(V_m \cup V_c)}$$ - a similarity by structure of reflected objects;

$$z_a(m,c) = \left(1 - \frac{\sum_{\forall a \in (A_m \cap A_c)} n(w_c(a))}{card(A_m \cap A_c)}\right) \cdot \frac{card(A_m \cap A_c)}{card(A_m \cup A_c)}$$ - a similarity by structure of links between objects;

$$z_t(m,c) = \left(1 - \frac{\sum_{\forall t \in (T_m \cap T_c)} n(w_c(t))}{card(T_m \cap T_c)}\right) \cdot \frac{card(T_m \cap T_c)}{card(T_m \cup T_c)}$$ - a similarity by multimedia textual annotations,

$n(x) = |0.1 \cdot \lg(1+x)|_{\mathrm{mod}1}$ - function for normalizing the weights;

$\lambda_v = 0, \text{if } card(V_m \cup V_c) = 0$; $\lambda_a = 0, \text{if } card(A_m \cup A_c) = 0$; $\lambda_t = 0, \text{if } card(T_m \cup T_c) = 0$.

To aggregate the similar multimedia in a class it is necessary to evaluate the distance between separate instances of multimedia $h(m_k, m_l)$. To solve this distance we will also taking into account a concordance of objects and links reflected in the multimedia m_k, m_l, and coincidence of keywords form their text annotations:

$$h(m_k,m_l) = 1 - z(m_k,m_l) = 1 - \frac{\lambda_v \cdot z_v(m_k,m_l) + \lambda_a \cdot z_a(m_k,m_l) + \lambda_t \cdot z_t(m_k,m_l)}{\lambda_v + \lambda_a + \lambda_t}, \quad (2)$$

where $z_v(m_k,m_l) = \frac{card(V_{m_k} \cap V_{m_l})}{card(V_{m_k} \cup V_{m_l})}$ - a similarity by structure of reflected objects,

$z_a(m_k,m_l) = \frac{card(A_{m_k} \cap A_{m_l})}{card(A_{m_k} \cup A_{m_l})}$ - a similarity by structure of links between objects,

$z_t(m_k,m_l) = \frac{card(T_{m_k} \cap T_{m_l})}{card(T_{m_k} \cup T_{m_l})}$ - a similarity by multimedia textual annotations,

$\lambda_v = 0, \text{if } card(V_{m_k} \cup V_{m_l}) = 0$; $\lambda_a = 0, \text{if } card(A_{m_k} \cup A_{m_l}) = 0$; $\lambda_t = 0, \text{if } card(T_{m_k} \cup T_{m_l}) = 0$.

To define the quality of partition of multimedia set by classes may be used the distance between multimedia classes $h(c_i, c_j)$. We will evaluate it in consideration of the class objects concordance, links between them, keywords and their weights:

$$h(c_i,c_j)=1-z(c_i,c_j)=1-\frac{\lambda_v \cdot z_v(c_i,c_j)+\lambda_a \cdot z_a(c_i,c_j)+\lambda_t \cdot z_t(c_i,c_j)}{\lambda_v+\lambda_a+\lambda_t}, \quad (3)$$

where $z_v(c_i,c_j)=\left(1-\frac{\sum_{\forall v\in(V_{c_i}\cap V_{c_j})} n\left(\left|w_{c_i}(v)-w_{c_j}(v)\right|\right)}{card(V_{c_i}\cap V_{c_j})}\right)\cdot\frac{card(V_{c_i}\cap V_{c_j})}{card(V_{c_i}\cup V_{c_j})}$ - a similarity by structure of reflected objects;

$$z_a(c_i,c_j)=\left(1-\frac{\sum_{\forall a\in(A_{c_i}\cap A_{c_j})} n\left(\left|w_{c_i}(a)-w_{c_j}(a)\right|\right)}{card(A_{c_i}\cap A_{c_j})}\right)\cdot\frac{card(A_{c_i}\cap A_{c_j})}{card(A_{c_i}\cup A_{c_j})}$$ - a similarity by structure of links;

$$z_t(c_i,c_j)=\left(1-\frac{\sum_{\forall t\in(T_{c_i}\cap T_{c_j})} n\left(\left|w_{c_i}(t)-w_{c_j}(t)\right|\right)}{card(T_{c_i}\cap T_{c_j})}\right)\cdot\frac{card(T_{c_i}\cap T_{c_j})}{card(T_{c_i}\cup T_{c_j})}$$ - a similarity by textual annotations;

$\lambda_v=0,\text{if } card(V_{c_i}\cup V_{c_j})=0$; $\lambda_a=0,\text{if } card(A_{c_i}\cup A_{c_j})=0$; $\lambda_t=0,\text{if } card(T_{c_i}\cup T_{c_j})=0$.

4. Multimedia Class Updating

For adding the multimedia ***m*** into the class ***c*** it is necessary to update the multimedia class model using the machine learning. During this operation the only weights vertices, edges and keywords of class, which is duplicated in the multimedia ***m*** should be increased in the class. According to this rule we should:

1) modify the objects' weights $w_c(v_i)$ from formula:

$\forall v_i \in V'=V_m \cap V_c \rightarrow w_c(v_i)=w_c(v_i)+1$;

2) modify the links' weights $w_c(a_i)$:

$\forall a_i \in A'=A_m \cap A_c \rightarrow w_c(a_i)=w_c(a_i)+1$;

3) modify the keywords' weights $w_c(t_i)$:

$\forall t_i \in T'=T_m \cap T_c \rightarrow w_c(t_i)=w_c(t_i)+1$;

4) add information about non-duplicated objects, links and keywords into the class ***c*** with the weight equal to 1:

$c=c+(m \setminus c)=\{V_c=V_c+V_m \setminus V_c, A_c=A_c+A_m \setminus A_c, T_c=T_c+T_m \setminus T_c\}$.

In case of a semantically closed class absence, the information about new multimedia can be added in the knowledge base as a new class with weights = 1.

5. Multimedia Clustering

For automatic multimedia classification and partition into the classes we propose to use an *agglomerative hierarchical method of clustering* [Pedrycz, 2005]. According to this method we will start clustering with unique clusters for each multimedia, and then unite these initial clusters into the new clusters and form two-level hierarchical structure, in which:

- **on first level:** clusters C_i with the minimum distance between clusters $h_{\min}=1$;
- **on second level:** subclusters C_{ij} with the $h_{\min}<1$.

To solve $h_{\min}$ you can use expression (2) or (3).

We will break up of multimedia into second level subclusters iteratively by changing h_{min} from 0 to 1, until the appropriate quality will be achieved. For evaluating this quality we suggest to measure the intrinsic homogeneity of a cluster η_o and the measure of clusters' heterogeneousness η_m:

$$\eta_{o_i} = \begin{cases} \frac{2}{N_i(N_i-1)} \sum_{j=1}^{N_i} \sum_{k=j+1}^{N_i} h(m_j, m_k), \text{if } N_i > 1 \\ 1, \text{otherwise} \end{cases}$$

, where N_i – the number of multimedia united in cluster C_i;

$$\eta_m = \begin{cases} \frac{2}{N_C(N_C-1)} \sum_{j=1}^{N_C} \sum_{k=j+1}^{N_C} h(m_c^{\langle C_j \rangle}, m_c^{\langle C_k \rangle}), \text{if } N_C > 1 \\ 0, \text{otherwise} \end{cases};$$

where $m_c^{\langle C_i \rangle}$ - central multimedia of cluster C_i, with minimum distances to other multimedia of a class: $\sum_j h(m_j, m_c) = \min, \forall m_j \in C_i$;

N_C – the number of parsed clusters;

$h(m_k, m_l)$ was defined in (2) above.

After clustering multimedia sets from each cluster $m_i \in C_j$ is aggregated into the classes c_i by using operation of multimedia class updating described in section 4 of this paper.

Conclusion

While multimedia documents producing and using is became a usual practice today, there is an unsolved problem of the semantic gap between multimedia content and its text descriptions defined by users. To solve this issue in this paper was presented the combined approach to multimedia indexing and retrieval which unite low level descriptors of multimedia and its high-level text annotations into one model. Such approach to presentation of multimedia semantics is based on aggregation of multimedia, that are similar by content and text annotation, into the classes, containing generalized descriptions of objects and links reflected in multimedia, and keywords from some thesaurus. Also we described the hierarchical clustering and machine learning operations which are used to create and update these classes.

Such approach should allow expanding a multimedia search and navigating area due to including of the media documents with a similar content and textual description.

Bibliography

[AAT] Getty's Art and Architecture Thesaurus. - http://www.getty.edu/research/conducting_research/vocabularies/aat/

[Dance et al., 1996] Dance S., Caelli T., Liu Zhi-Qiang. Picture Interpretation: A Symbolic Approach//World Scientific Series In Machine Perception And Artificial Intelligence; Vol. 20 - 1996.

[Duran et al., 1974] Benjamin S. Duran, Patrick L. Odell. Cluster analysis. A survey. - Springer-Verlag. Berlin-Heidelberg-New York, 1974.

[Goodrum, 2000] Goodrum A. A. Image information retrieval: An overview of current research // Informing Science, 3(2):P.63-66, February 2000.

[LCTGM] Thesaurus for Graphic Materials // Library of Congress. - http://www.loc.gov/rr/print/tgm1/

[MPEG-7] MPEG-7 Overview. <http://www.mpeg-7.com> (Industry Focus Group)

[Ozkarahan, 1986] Ozkarahan E. Database machines and database management. - Prentice-Hall, Inc, 1986. - 636p.

[Pedrycz, 2005] Pedrycz W. Knowledge-based clustering: From Data to Information Granules. - John Wiley & Sons, 2005. - 316p.

[Petridis et al., 2004] Petridis K., Kompatsiaris I., Strintzis M.G., Bloehdorn S., Handschuh S., Staab S., Simou N., Tzouvaras V., Avrithis Y. Knowledge representation for semantic multimedia. Content analysis and reasoning. - EWIMT, 2004.

[Petrushin and Khan, 2007] Petrushin V. A. and Khan L. (Eds). Multimedia Data Mining and Knowledge Discovery. - Springer-Verlag London Limited, 2007. – 521 p.

[Stamou and Kollias, 2005] Stamou G. and Kollias S. (Eds). Multimedia Content and the Semantic Web. - John Wiley & Sons Ltd, 2005 - 392 p.

[Vihrovsky and Ignatenko, 2006] Vihrovsky K, Ignatenko A. Using MPEG-7 for visual data classification and retrieval//Internet magazine "the Drawing and Multimedia". - http://cgm.graphicon.ru/content/view/161/61/

Authors' Information

Dmytro Nochevnov *– Associate professor, "Information Technologies of Design" department of Cherkasy State Technological University, Cherkasy, Ukraine, 18006.*

Major Fields of Scientific Research: Data Mining and Knowledge Discovery, Education on Information Technologies.

E-mail: dmitry.ndp@gmail.com.

THE INTEGRATIVE STRUCTURE OF MULTIMEDIA COURSEWARE OF ELECTROTECHNICAL LECTURE COURSE

Larisa Zaynutdinova, Natalia Semyenova

Abstract: *The article deals with necessity and value of lecture courses in Higher Education Institutions, with animation of educational and cognitive activity of the students during the lecture course with multimedia courseware (MC) application, and with the integrative structure of multimedia courseware of electro-technical lecture course.*

Key words: *Education, multimedia lecture, multimedia-based courseware*

ACM Classification Keywords*: K.3.1 Computer Uses in Education, J.2 Physical Sciences and Engineering.*

Introduction

Lecture in a Higher Education Institution is a principal element of didactical cycle, aimed at development of target background for the students' further studying the material. The word "lecture" originates from the Latin word "lectio", literally meaning "reading". Lectures first appeared in ancient Greece and were further developed in ancient Rome. In Russia a great contribution into the development of lecturing as a form of education was made by M.V. Lomonosov, the founder of the first Russian University.

The forms and tasks of lectures were changing and improving together with development of Higher Education. In the Middle Ages, when there were no manuals, and lectures were the only source of knowledge for the students, the professors, standing at the lectern, were either fast or slowly reading their material. Since the 18th century lecture turned into the professor's oral narration. Nowadays lecture is no longer a reading of material or retelling a textbook, it is "an original investigation, an individual analyzing and synthesizing, considering of what one has seen with one's own eyes and apprehended with one's mind" [Ножин, 1983] . Previously being a monologue, "it is getting more and more a form of collective thinking aloud of both the lecturer and the students" [Фейгенберг, 1989] .

The essence of lecture was well shaped by Russian scientist P.A. Florensky: "The essence of lecture is spontaneous scientific life, it is considering scientific objects together with students, it is not issuing ready-made, stereotyped conclusions out of stores of chairborne scholarism. Lecture is introducing students into the process of scientific work, involving them into scientific creation, it is a sort of demonstrative and even experimental method-teaching and not just transferring "truisms" of science in its "modern" state...A lecture… must not teach this or that number of facts, conclusions or theories, it must teach how to work, form taste to scientific vision, give "catalizer", yeast of intellectual activity".

In the curriculums of Higher Education Institutions of Russian Federation lectures take 40-50% of academic time. This form of education is a guideline for the other forms of studying. The material, given to the students at the lectures becomes a basis for their further educational and cognitive activity at seminars and laboratory classes. In spite of a great role the lectures have in the educational process many teachers report on the students' passivity and low level of their educational and cognitive activity at the lectures. That is why they suggest changing hours given for lectures with hours on individual work.

We can not agree on this point as:

- It is lecture that encourages students to developing necessity and motivation to learn the subject or the topic;
- It is lecture that enables transferring a great volume of information in a rational way, getting the students acquainted with the theory of the subjects;
- It is lecture that gives the scientific material directly, introducing the students into the lecturer's scientific laboratory, shows the lecturer's personal attitude to the subject, and besides students' claims and level of education are taken into account;
- It is lecture that develops a scientific outlook, gives examples and guidelines for individual independent work, trains in classification of notional units and their relation to methodological, theoretical and actual level of knowledge.

So, we believe that lecture still should remain both a leading method of teaching and a principle form of organizing educational process in the Higher Education Institutions, while animation of educational and cognitive activity of the students can be achieved by applying information technologies [Зайнутдинова, 1999] and unconventional lectures. Multimedia lecture is one of them, and it is read with the help of multimedia courseware.

Multimedia Courseware (MC) of Electrotechnical Lecture Course

We call Multimedia Courseware (MC) of Electrotechnical Lecture Course a complex of interconnected educational software (informational, training, modeling, reference and encyclopedic, checking-up) providing a complete structure of educational and cognitive activity: purpose, motive, the activity itself, result - on condition of interactive feedback, carried out according to multimedia techniques [Семенова, 2007].

Using this definition under MC of Electrotechnical Lecture Course we recognize a MC with dominating informational component.

To animate educational and cognitive activity of the students at the multimedia lectures we have elaborated an integrative structure MC (LC) comprising blocks of educational material putting to practice the lecture's didactic components and its principal functions (fig.1).

According to the primary structural components of the educational and cognitive activity [Харламов, 2005], we distinguish between the following didactic components of a multimedia lecture: **objective, demand-motivational, conceptual, operative-active, emotional-volitive, control-regulational, evaluative-resultative.**

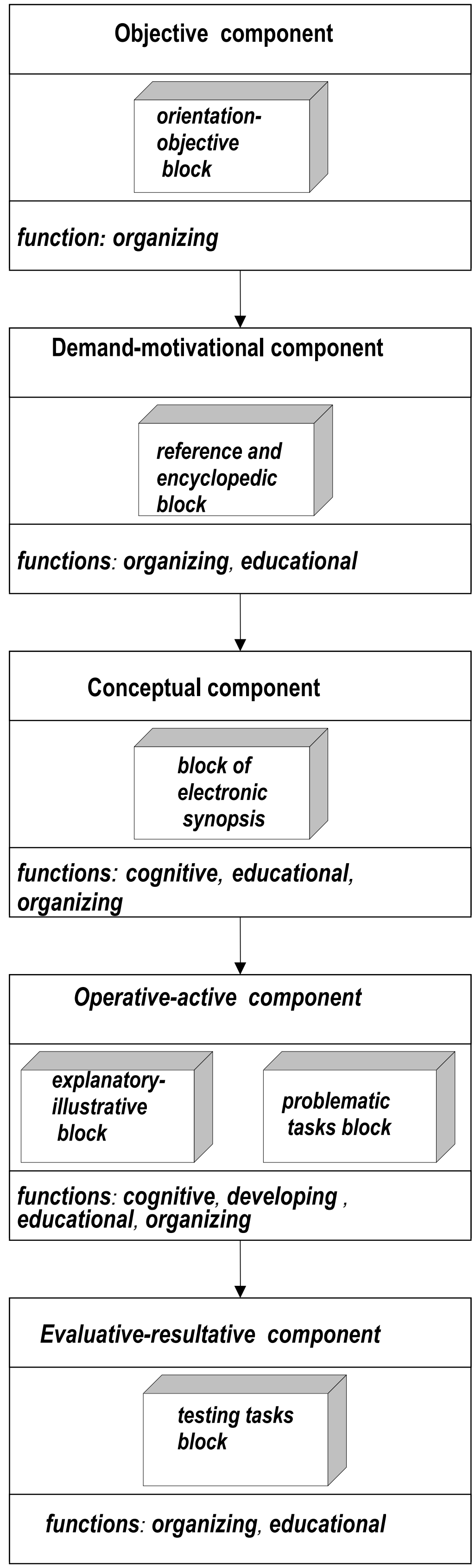


Fig.1. The Integrative Structure of Multimedia Courseware of Lecture Course

These components can be briefly characterized as follows:

The **objective component** includes the theme, purposes and tasks of the lecture, the link between this lecture with the previous and the following ones, as well as connections of the theme with other subjects and sciences, studied by the students at the senior courses.

The **demand-motivational component of the multimedia lecture.** The principal task of this structural component is to get the students interested into the theme of the lecture. If the task of objective structural component is easy to realize at both a traditional lecture and at a multimedia lecture, the main task of the demand-motivational component can be better realized when using Multimedia Courseware. To attract the students' interest to the theme of the lecture with the help of MC a lecturer can begin it with historical documents on the scientific contribution of the scientists into the development of the theme, or give some heuristic tasks faced by the electrotechnicians on the problem of the lecture. While considering and solving such tasks at the beginning of the lecture the students get a bit of motivating intellectual warming-up, stimulating their interest to the topic and programming them for effective work during the lecture.

The **conceptual component** is one of the principal didactic components of a lecture on any subject of the State Educational Standard, fixed in text-books, manuals, copyrighted lectures, including those in electronic form.

The content of the lecture first of all should be scientific and reflect modern tendencies of investigating the subject. As soon as an electronic synopsis can be easily amended and changed, it has an undeniable advantage as compared to a printed material. Moreover, the electronic synopsis allows shortening the time between its composing and issuing.

The main task of **operative-active component** is to organize the educational and cognitive activity of the students at the lectures, to facilitate perception of material as well as considering and effective memorizing information.

The **operative-active component** at a multimedia lecture can be magnified via visualization of the material and possibility to enhance the lecture with programs of imitative modeling, able to create problematic situations.

The **evaluative-resultative component** is a new didactic component of a lecture, substantially amending methods of reading multimedia lectures in comparison with traditional ones. This component provides the feedback and the students' self-control as well as correction of the reading methods by the professor.

The evaluative-resultative component at multimedia lectures is realized due to express-testing, which allows the students to self-check their knowledge while comparing their results with the reference material. The introduction of this component turns the multimedia lecture into a lecture of closed type of managing the educational activity.

The didactic and educational purposes of lecture are (according to [Виленский, 2004]:

- Giving to students modern, complete, interconnected knowledge with the level defined by the objective goal to every certain theme;
- Providing creative activity of the students together with the lecturer during the lecture ;
- Developing professional and business qualities of the students, stimulating their interest to the subject and shaping their independent and creative way of thinking.

According to the above mentioned purposes of the lecture, M.Y. Vilensky [Виленский, 2004] distinguishes between the following functions of lecture: cognitive, developing, educative and organizing.

The **cognitive function** of lecture provides students with fundamental notions of the science and defines scientifically based ways of solving its practical tasks and problems. The **developing function** of lecture is in orienting students to rely on their thinking, not on their memory while studying. The cognitive function corresponds to reproductive level of educational and cognitive activity, while the developing function corresponds to the productive one. The **educational function** is realized in case the lecture includes not only actual knowledge in this or that professional sphere but also idealogic, general scientific and humanitarian information. The **organizing function** of lecture is in managing the independent work both during the lecture and while self-preparing.

All these functions are true for multimedia lectures, moreover they become even more valid due to a higher level of program and psycho-pedagogical potential of MC (LC).

In philosophy structure is a form of the content existence. So, the structure of Multimedia Courseware should be a combination of blocks of the educational material in this or that sphere. We distinguish between the following blocks: **orientation-objective** block, **reference and encyclopedic** block, the block of **electronic synopsis**, **explanatory-illustrative** block, the block of **problematic tasks** and that of **testing tasks**.

Here are brief characteristics of the blocks.

The **orientation-objective** block corresponds to the objective didactic component of multimedia lecture. The material of the block is structurally represented according to the functional purpose of the objective component and it includes the themes of the lectures within the program of the course, their aims and tasks; skills, types and ways of activities for the students to master.

As soon as the educational process is first of all connected with its clear target-setting and the students' acceptance and understanding of these targets, the primary function of orientation-objective block is organizing one.

The **reference and encyclopedic** block is responsible for demand-motivational component of MC due to the included biographical data and main achievements of outstanding scientists of the sphere; information on results of new scientific achievements and prospects of further development; principal notions and definitions on the subject according to the State Educational Standard.

Students' demand for knowledge and their interest to it are influenced by a great number of pedagogical methods, devices and other factors. I.F. Kharlamov [Харламов, 2005] notes, that the teacher as a person, his outlook, his erudition and experience are very effective in this aspect. If a teacher profoundly knows his subject, while teaching he or she users interesting facts and details, impressing students with the range of his or her outlook and charming them with his or her educational level. In this case following psychological mechanism to copy, students come to the inner conflict between their level of knowledge and the desirable example, and wishing to cope with the disharmony they get a push to a more active learning.

The information of this block contributes to the students' interests to the subject, stimulates their desire to study, develops their scientific outlook, and greatly influences their personal growth.

The block of **electronic synopsis** corresponds to conceptual component, representing a text synopsis of lectures with explanations to every slide of the explanatory-illustrative block. The block of electronic synopsis can be used both by lecturers while preparing for multimedia lectures and by students of all forms of studying while reading for seminars. The main functions of the block are: cognitive, organizing and educational.

The **explanatory-illustrative** block provides reproductive level of educational and cognitive activity of the students, it is used while giving the material through explanations and illustrations. The block includes multimedia lectures in modules – i.e. structurized by topics. Every module contains the title of the theme, educational material in slides to be shown by the lecturer (photos, electric circuit diagrams, charts, schemes). While composing slides for the multimedia lectures the teacher should follow ergonomic requirements. The main functions of this block are: cognitive, developing, organizing and educational.

The block of **problematic tasks** manages the productive level of educational and cognitive activity of the students, providing problem-solving presentation of material. It is represented by the complex of problematic tasks, grouped thematically. Every problematic task is a computer-visualized data of the problem to be solved with theoretically possible decisions, provided by the lecturer. The latter uses it via programs of imitating modeling. The main functions of this block are: cognitive, developing, organizing and educational.

Both blocks – the explanatory-illustrative and that of problematic tasks realize the operative-active component of MC.

The block of testing tasks serves for express-testing at multimedia lectures and represents a number of tests to every theme. Express-testing at the end of multimedia lectures makes educational-cognitive activity of the students conscious and instills self-discipline in students.

MC of lecture course in the subject “Theoretical backgrounds of electrotechnics” was created in the HTML-format and registered in the Brunch Collection of Algorithms and Programs (Moscow). The orientation-objective, reference and encyclopedic blocks and that of electronic synopsis are made in **Word** in the form of hypertext. The explanatory-illustrative block as well as blocks of problematic tasks and tests are created with the help of **Micromedia Flash** and **Power Point** programs, with their animating and sound effects, possibilities to insert photos, video clips, fragments of imitating modeling in **Electronics Workbench** and **Matlab**. The lecturer regulates reproduction of this or that animated picture by clicking the mouse pointer.

Conclusion

The MC lectures turned out to be visually dynamic, convincing, emotional, colorful. The results of the questionnaire survey held among the students and joint scientific investigation of Multimedia Technologies laboratory and The Health Centre of Orenburg State University proved that the volume and quality of the retention of material by the students considerably grows at MC lectures, motivation to study the subject appears, educational and cognitive activity becomes more vivid.

Bibliography

1. Nozhin, E.A. “The Logics of Presentation – the Logics of Convincing”// “Agitator”, 1983, № 21.
2. Feigenberg, I.M. “The Lecture Compatible with the Requirements of Time ”/ I.M. Feigenberg, // “Vestnik Vysshey Shkoly” – 1989, № 1, pp 33-36.
3. Zaynutdinova, L. H., “Producing and Appliance of Electronic Manuals” (illustrated by technical sciences) : monograph/ L. H. Zaynutdinova. – Astrakhan : CNTEP, 1999. – 364p.
4. Semyenova, N.G. “Theoretical Backgrounds of Producing and Appliance of Multimedia Courseware of Electronic Lecture Courses”. Monograph/ N.G. Semyenova,. – Orenburg, IPF “Vestnik”, 2007. – 317p.
5. Kharlamov, I.F. Pedagogika/ I.F. Kharlamov, the 4th Edition, - M.: Gardariki, 2005. – 520 p.
6. Vilenskiy, V.Y. “Techniques of Profession-Oriented Education in the Higher Education Institutions”/ V.Y. Vilenskiy, P.I. Obraztsov, A.I. Uman. – M.: Pedagogicheskoye Obtchestvo Rossii, 2004. – 192p.

Authors' Information

Zaynutdinova Larisa Hasanovna, *doctor of pedagogical science, professor,*

Astrakhan State University, Russia

414025, г. Астрахань, ул.Татищева, 20 а; e-mail: lzain@mail.ru,

Major Fields of Scientific Research: Information Technologies in Education, Electrical Engineering

Semyenova Natalia Gennadievna, *doctor of pedagogical science,*

Orenburg State University, Russia

460018, г.Оренбург, пр. Победы, 13; e-mail: tomsk@house.osu.ru

Major Fields of Scientific Research: Information Technologies in Education, Electrical Engineering

ITHEA INTERNATIONAL SCIENTIFIC SOCIETY

The ITHEA International Scientific Society (**ITHEA ISS**) a successor of the international scientific co-operation organized within 1986-1992 by international workgroups (**IWG**) researching the problems of data bases and artificial intelligence. As a result of tight relation between these problems in 1990 in Budapest appeared the international scientific group of Data Base Intellectualization (**IWGDBI**) integrating the possibilities of databases with the creative process support tools. The leaders of the IWGDBI were Prof. Victor Gladun (Ukraine) and Prof. Rumyana Kirkova (Bulgaria). Starting from 1992 till now the international scientific co-operation has been organized by the Association of Developers and Users of Intellectualized Systems (**ADUIS**), Ukraine. It has played a significant role for uniting the scientific community working in the area of the artificial intelligence.

To extend the possibilities for international scientific collaboration in all directions of informatics by wide range of concrete activities, in 2002 year, the Institute for Information Theories and Applications FOI ITHEA (**ITHEA**) has been established as an international nongovernmental organization.

ITHEA Scientific Council includes:

Honorable Chair Persons: Victor Gladun (Ukraine) and Rumyana Kirkova (Bulgaria);

Chairman: Levon Aslanyan (Armenia);

Members: Adil Timofeev (Russia), Alexander Kleshchev (Russia), Alexander Kuzemin (Ukraine), Alexander Palagin (Ukraine), Alexey Voloshyn (Ukraine), Arkadij Zakrevskij (Belarus), Constantin Gaindric (Moldova), Hasmik Sahakyan (Armenia), Ilia Mitov (Bulgaria), Juan Castellanos (Spain), Koen Vanhoof (Belgie), Krassimir Markov (Bulgaria), Larisa Zainutdinova (Russia), Laura Ciocoiu (Romania), Luis Fernando de Mingo (Spain), Martin Mintchev (Canada), Nikolay Zagoruiko (Russia), Peter Stanchev (USA), Stefan Karastanev (Bulgaria), Tatyana Gavrilova (Russia), Vladimir Lovitskii (GB), Vladimir Ryazanov (Russia).

ITHEA Scientific Sections: Business Informatics (Koen Vanhoof), Cooperative Scientific Projects (Hasmik Sahakyan), Decision Support (Alexey Voloshyn), Information Interaction (Stefan Karastanev). Intelligent Engineering Systems (Martin Mintchev), Logical Inference and Design (Arkadij Zakrevskij), Modern (e-) Learning (Larisa Zaynutdinova), Natural Computing (Juan Castellanos), Pattern Recognition (Vladimir Ryazanov), Philosophy and Methodology of Informatics (Krassimir Markov), Scientific Innovation Management (Luis Fernando de Mingo), Software Engineering (Ilia Mitov), Theoretical Informatics (Levon Aslanyan).

ITHEA is aimed to support international scientific research through international scientific projects, workshops, conferences, journals, book series, etc. The achieved results are remarkable. The ITHEA became world-wide known scientific organization. One of the main activities of the ITHEA is building an International Scientific Society aimed to unite researches from all over the world who are working in the area of informatics.

Now, the **ITHEA International Scientific Society** is joined by more than **2200** members from ***47*** countries all over the world: *Armenia, Azerbaijan, Belarus, Brazil, Belgium, Bulgaria, Canada, China, Czech Republic, Denmark, Egypt, Estonia, Finland, France, Germany, Greece, Hungary, India, Iran, Ireland, Israel, Italy, Japan, Jordan, Kyrgyz Republic, Latvia, Lithuania, Malaysia, Malta, Mexico, Moldova, Netherlands, Poland, Portugal, Romania, Russia, Scotland, Senegal, Serbia and Montenegro, Sweden, Spain, Sultanate of Oman, Turkey, UK, Ukraine, and USA.*

ADUIS

ASSOCIATION OF DEVELOPERS AND USERS OF INTELLIGENT SYSTEMS

ADUIS consists of about one hundred members including ten collective members. The Association was founded in Ukraine in 1992. The main aim of ***ADUIS*** is to contribute to the development and application of the artificial intelligence methods and techniques. The efforts of scientists engaged in ***ADUIS*** are concentrated on the following problems: expert system design; knowledge engineering; knowledge discovery; planning and decision making systems; cognitive models designing; human-computer interaction; natural language processing; methodological and philosophical foundations of AI.

Association has long-term experience in collaboration with teams, working in different fields of ***research and development***. Methods and programs created in Association were used for revealing regularities, which characterize chemical compounds and materials with desired properties. Some thousands of high precise prognoses have been done in collaboration with chemists and material scientists of Russia and USA.

Association can help ***businessmen*** to find out conditions for successful investment taking into account region or field peculiarities as well as to reveal user's requirements on technical characteristics of products being sold or manufactured.

Physicians can be equipped with systems, which help in diagnosing or choosing treatment methods, in forming multi-parametric models that characterize health state of population in different regions or social groups.

Sociologists, politicians, managers can obtain the Association's help in creating generalized multi-parametric "portraits" of social groups, regions, enterprise groups. Such "portraits" can be used for prognostication of voting results, progress trends, and different consequences of decision making as well.

Association provides a useful guide in technical diagnostics, ecology, geology, and genetics.

ADUIS has at hand a broad range of high-efficiency original methods and program tools for solving analytical problems, such as knowledge discovery, classification, diagnostics, and prognostication.

ADUIS unites the creative potential of highly skilled scientists and engineers

Since 1992 ***ADUIS*** holds regular conferences and workshops with wide participation of specialists in AI and users of intelligent systems. The proceedings of the conferences and workshops are published in scientific journals. ***ADUIS*** cooperates through its foreign members with organizations that work on AI problems in Russia, Belarus, Moldova, Georgia, Bulgaria, Czech Republic, Germany, Great Britain, Hungary, Poland, etc. ***ADUIS*** is the collective member of the European Coordinating Committee for Artificial Intelligence (ECCAI).

www.aduis.com.ua

For contacts: V.M.Glushkov Institute of Cybernetics; National Academy of Science of Ukraine;

Prospect Akademika Glushkova, 40, 03680 GSP Kiev-187, Ukraine;

Phone: (380+44) 5262260; Fax: (380+44) 5263348; E-mail: glad@aduis.kiev.ua

TABLE OF CONTENTS OF VOLUME 3, NUMBER 1