ITHEA

International Journal

INFORMATION TECHNOLOGIES & KNOWLEDGE

2009 Volume 3 Number 4

International Journal
INFORMATION TECHNOLOGIES & KNOWLEDGE

Volume 3 / 2009, Number 4

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 4, 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)**

HIERARCHICAL THREE-LEVEL ONTOLOGY FOR TEXT PROCESSING

Victor Gladun, Vitalii Velychko, Leonid Svyatogor

Abstract: *The principal feature of ontology, which is developed for a text processing, is wider knowledge representation of an external world due to introduction of three-level hierarchy. It allows improving semantic interpretation of natural language texts.*

Keywords: *ontology, text processing, thematic text analysis.*

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

Introduction

One of the practical purposes of an artificial intelligence methods use is *Text Processing* - the semantic analysis of naturally-language texts both general thematic orientations and concerning to various domains [1, 2, 3, 4]. Language thesauruses and linguistic ontologies are developed for this purpose and set the certain conceptually-expressed system of knowledge. Distinction between thesauruses and linguistic ontologies consists in volume knowledge representation and methods of classification (structurization) the conceptual environment. Among significant achievements in the field of lexicographical representation of natural languages it is possible to specify: Roget P.M. and Dornzaif F. thesaurus [5], the Ideographic dictionary of the Russian language [6], the Thesaurus of the Russian language *RuThes* [1], resource *WordNet* [7]. However, ontology is more adequate to the conceptual texts analysis, because by means of ontology connections between objects, concepts and its properties can be most full presented by all of them a variety.

As is known, ontology in a general view represents the sets of the domain terms, relations between terms and the domain interpretation functions on the terms and the relations. It is possible to point out the next advantages of ontologies: a) deep interaction between both described objects and phenomena and the contextual environment; b) the economical storage of the information demanding storing domain terms and relations, instead of stages memorizing; c) universal character of ontology, which allows within its structure to solve both problem – knowledge synthesis and analysis; d) "flexibility" of the knowledge structure which is adjusted on diverse domains.

Ontologies are developing, more often, for concrete domains ("narrow ontologies"), at that process begins with the analysis of a text collection [1, 2]. However, in a context of the given work the principal interest have linguistic ontologies ("wide ontologies"), which cover the knowledge continuum in different spheres of human activity, so as *Mikrokosmos* [8], *SUMO* [9] *and Knowledge Representation* by J. Sowa [10].

Two Approaches to Creation of Ontologies. Topicality of an Compromise

Now two opposite approaches to construction of the linguistic ontologies were formed. First of them – *"search ontologies"* – is purposeful on problems of automatic text processing in various domains. For example, Sociopolitical thesaurus is focused on the social and political life and used in such applications of automatic text processing as conceptual indexing, automatic text categorization [1]. The ontology synthesis technology is based on the analysis of the representative text collection (sometimes – tens thousands sources). Sizeable difficulties are in the fact that is impossible "to fish out" the *context knowledge* out of professional text collections, what is quite necessary to texts understanding. Therefore, it requires substantial experts' efforts for completion of the ontology by complementary concepts and to organization semantic relationships between domain and external world. Moreover, in many cases the semantic communications do not lend its to formalization. As a result, it is impossible to avoid of essential intervention of expert. The final structure becomes too bulky in use and difficult for tuning.

On the other hand, there is an opposite approach. As a starting point of construction "abstract ontologies" consider *Universe* [10], *Essence* [9], *All* [8]. The knowledge representation is set by the branched out hierarchical structure. Concepts ontology reaches the maximal generalization and abstraction, and on this height, more often is remaining. By virtue of it, application of universal ontology to concrete texts analysis seems to be very problematic. Besides that, the formal relations used in abstract ontologies far not always describe the properties of real world in its reflection by man. Therefore, some authors introduce into consideration "flexible" relations: "*conceptual dependence relations*" [11], "*role relations*" [1], "*symmetric and asymmetrical associations*" [2] or "*associative relations*" [12]. Using of similar relations to thematic analysis NL text seems very constructive.

Thus, in a context of the lead analysis, the problem consists in separation between professional-focused and abstract ontologies, i.e. – between a particular description and abstract presentation of knowledge. Resources, which are available in a free access, do not allow realizing thematic text analysis in Russian. The way out from this opposition consist, as it seems, in synthesis "wide" and "narrow" ontologies. It is necessary to create an integrated structure, where it can be distributed and balanced described both the general, meaningful knowledge about nature and society, and concrete domain resources. Such structure can reflect a hierarchical picture of the whole world.

The purpose of the given work consists in offering the concept of integrated hierarchical (three-level) representation model of environment, which: a) in the compressed kind and with a different degree of generalization (or detailed elaboration) reflects actual knowledge about structure of an external world; b) is focused on text processing both the general subjects and separate areas of knowledge and c) allows to integrate a professional knowledge into a conceptual network without reorganization of the upper and middle ontology levels.

This conception develops the *semantic (thematic) analysis method* of NL text processing by creation of the document synopsis. Procedure of making a synopsis is based on disclosing of the given theme by means of sequence of keys words, which are automatically generated with the program "KONSPEKT" [13]. However, defect of this method was in strongly simplified one-level model of the real world representation, which was not taken into account hierarchy and depth of external world.

Substantiation of the Approach to Construction of Hierarchical Ontology

For construction of multilevel ontology, the methods and mathematical models that contain models of ontology, knowledge and domain are used [4]. Offered here three-level associations ontology is intended for the decision of more specific problem - the thematic texts analysis. It defines a number of preconditions and features.

1. *Gnosiological conceptions in ontologies. Paradigm acceptance*

There are many approaching methods to problem of universe representation in philosophy, natural sciences and linguistics. The authors of different universal ontologies describe a world with such general categories as: *Essence = Material, Abstract* [3, 9]; *All = Object, Event, Property* [3, 8]; *Universum* (is divided on seven components) [3, 10]. That is not exclude and others, exotic variants: for example, classification of *All* into *Goodness* and *Evilness* may be successful... The question of substantiation usually is not considered.

However, more practical and pragmatic methods recommend the scientific methodology of the system analysis. It operates the following types of resources: *Substance; Energy; Information; Man; Organisation; Space; Time* [14].

These categories, in our opinion, possess a necessary diversity and they are objects of researches in physical and social sciences. This world outlook may be taken as a base of description of the external world.

However, the most modern is the materialistic idea proposed by academician V.I. Vernadsky. In accordance with it, all, what is just known, may be divided on two fundamental categories: *Inert substance* and *Alive substance*. The *Alive substance* is realized in *Biosphere* and *Noosphere* - sphere of Human activity. Both categories characterize the *Matter* fundamentally. This materialistic paradigm is put in a basis of offered ontology. It works out in detail in partition "Choice of categories for upper level of ontology".

2. *A choice of the general structure*

In correspondence with hierarchical picture of the knowledge about the world, we are distinguishing three levels in ontology construction. On the upper level are the general categories of universe; here the strict taxonomy is possible. Upper level summarizes the concepts of general knowledge and reflects the ontological basis.

The middle level is disclosing the base terms in more detail - by concepts having lesser level of commonality. Concepts, used on middle level, reflect the universal, popular and well-known terms existing in nature, society and environment; there are presented relations between them also.

At last, on the lower level is concentrated knowledge, which characterizes concrete situations and describe some environment. At this level is presented knowledge of the problem-focused area. Therefore, on this level two intersected blocks exist: one of them contains the concepts and words of common usage, which exist in general, and interdisciplinary texts; other block serves the domains. Due to this two-blocks structure the ontology may grow at the expense of new domains. In principle, the domain block may be empty - the working capacity of ontology is remained fully.

Corresponding to hierarchical structure of ontology we will use the term **HiO**.

3. *A choice of connections*

At upper level **HiO** the formal connection of type «the whole - a part» is applied. At middle and upper levels, except the formal connections, following specific types of connections are entered into consideration:

a) **A** reveals through **B**; **B** explains **A**;

b) **A** is characterized by property **E**;

c) **A** is *associatively* connected with **C** .

Widely used in **HiO** the term "associative connection" is not formal. It is necessary to reflect individual semantic correlation of two (any) concepts if it takes place. Associations have a situational and dynamical character. At an *information level* they can be able to open unexpected properties (or laws) of some object (or phenomena). At a *functional level,* they fix some dependence. At a *cognitive level* associative connection of two concepts means that one image has excited another. At a *logic level* association are predicative and implicative, but in most cases – not transitive.

Constructive Properties of Associations Ontology

Based on a practical orientation of the text processing, we shall specify some functional **HiO** features.

1. ***Completeness and taxonomy*** on upper level of the ontology signify that the chosen categories in aggregate are representing the Matter in the exhaustive way. Outside these categories, there should be no manifestation of the reality. Categories are subordinated to strict hierarchy and classification that excludes logic ambiguity of concepts.

2. ***Natural-scientific lexicon.*** The ontology categories and concepts should be common, simple and clear. They lexically expressed by those concepts and terms which were established in sphere of the general

knowledge, in natural and social sciences, in the socio-cultural environment. The upper level **HiO** can be supplemented with a special terms.

3. ***Connectivity on association***. Connections between concepts inside of levels and between ones include both formal and informal (associative) connections. Associative connections are actively used at middle and upper levels of **HiO** to fuller description of theme.

4. ***Antagonism reflection***. Concepts, reflecting properties, which have (inside of the given measure) its contrast, can be designated by pair's words (antonyms).

Synthesis of Hierarchical Associations Ontology

On the base of the formulated preconditions there is clear the following prospect of actions. It is necessary to choose categories, concepts and connections between them and distribute on the upper, middle and lower levels of hierarchical associations ontology. Thus on upper level the simplified strict model of the world is presented. On the middle level, it is necessary to disclose the categories of upper level, using wide concepts of interdisciplinary dialogue. On the lower level, the conceptual basis of middle level is to be described in detail. In addition, here professional domains knowledge is localized. Received three-levels network ontology should be later connected to linguistic database and integrated in the working system of NL texts analysis. On the Fig. 1 general block diagram of three-level hierarchical associations ontology is presented.

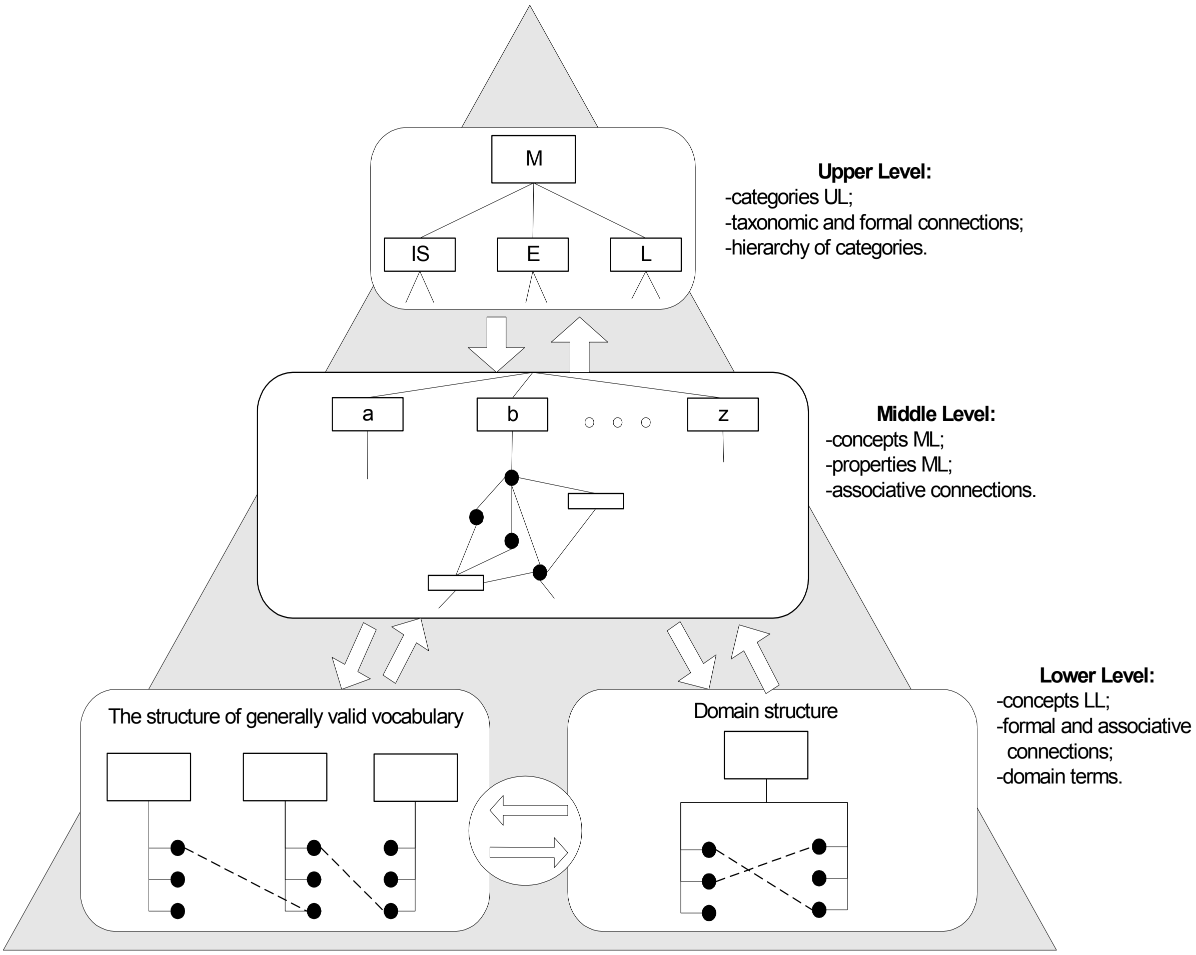


Fig1. The block diagram of three-level associations ontology.

Choice of Categories for Upper Level of Ontology

As it was specified above, within the limits of offered ontology the general picture of a Universe is subordinated to materialistic idea by V.I. Vernadsky. Apex of hierarchy is the philosophical category the *Matter*. It shows itself in exhaustive manner as *Inert* and *Alive substance*. On the other hand, the Inert substance may become the forms either *Substance* or *Energy* (which are passing each another under the Preservation Law). Therefore, we realize trichoyomy of the *Matter* on: *Substance* (inert), *Energy* and *Life* (the substance of *Alive*). In this case dividing is made on base "The Form existence of Matter". Each of three categories is presented by a number of subcategories, as is shown on Fig.2.

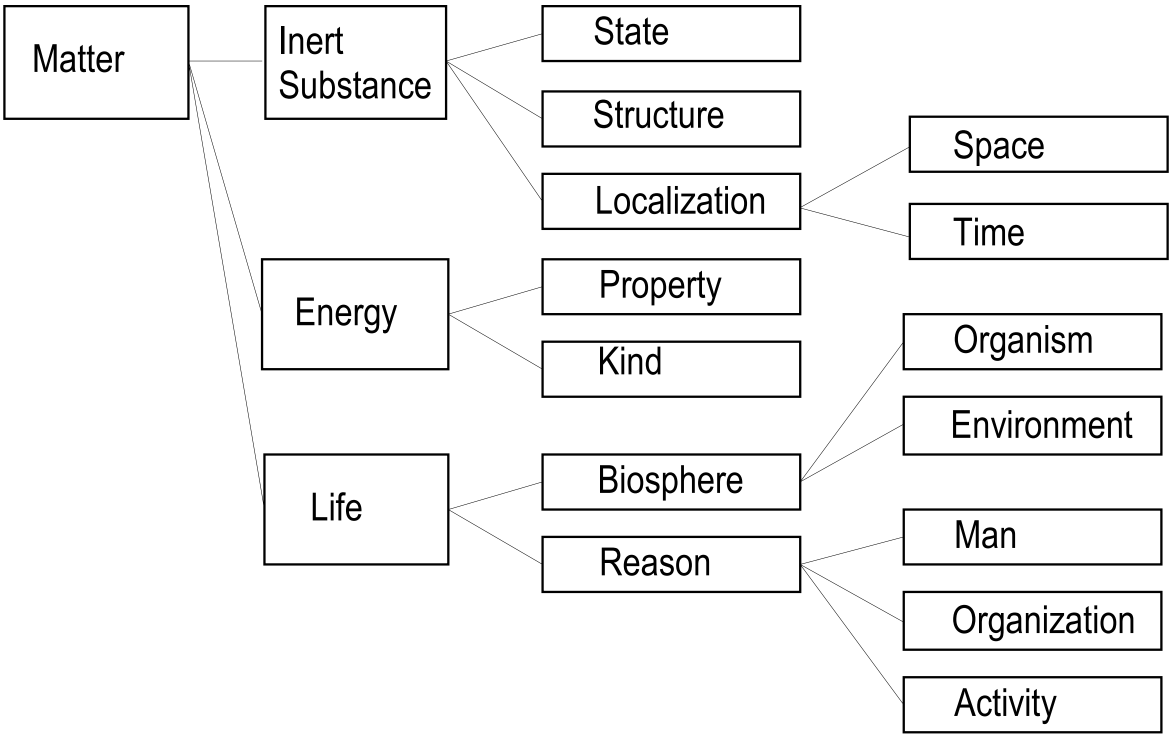


Fig. 2. Structure of top level ontology

It is necessary to add some explanations to the resulted ontological structure of upper level.

The first. At a pyramid of upper level, there are no such general categories of knowledge, as, for example: the Being, the Consciousness, the Measure, the State, the Property, the Quantity, the Quality and others. In **HiO** some of them are transferred on the middle level, owing to what they are released from philosophical sense and "work" as a terms of natural sciences.

The second. It is possible to show, that **HiO** upper level possesses property of completeness of conceptual volume. Really: a) *Substance, Energy* and *Life* are *the forms of the Matter realization*; b) the *Space* and *Time* serve as *the forms of the Matter distribution* and c) the *Reason* is *the way of the Matter reflection*. Summary these three metacategories are exhausting a metacategory "Being of the Matter". If to accept the given statement for an axiom, **HiO** covers all known (or real) properties of the Matter. As a result, the given categorical system on upper level is complete and closed.

In view of proposed above universal categories, concepts and comments the continuation of the ontological scheme may be formed.

The Principles Building the Middle Level of Ontology

Purpose of the middle level of ontology (MLO) is, on the one hand, to disclose all categories of upper level - to give them semantic filling, and with another – to form a semantic environment for the concordance with the lower level's concepts.

MLO represents such level of knowledge, which is common to a various areas; that is interdisciplinary knowledge. Per se, the middle level of hierarchy fixes itself a layer of valid human knowledge, which is generalized by collective experience in science, culture, practice – out of professional sphere. It operates by generally accepted words. A material to this level is formed by the knowledge engineer. The middle level is "conservative": it is a "constant" **HiO**s component. At the given level informal (associative) connections of type "object – property", which (in opinion of the expert) bear the helpful information for disclosing internal structure of ontology, are actively used. It is necessary to emphasize, that occurrence doubtful connections is not lack of associations ontology, quite the contrary – they open an opportunity to additional adjustment.

Brief description MLO. Middle level of ontology represents set of network structures: as a name (and initial node) of each structure serves a category of upper level; internal nodes are the concepts of the middle level; internal connections between concepts disclose the important characteristical properties of category to be done.

Here, with a view of place economy, concepts and the full structures making middle level of **HiO** are not presented. However, as an example, the structure of the *Organization* category from the *Reason* cluster (see Fig. 3) is shown.

The note. The interpretating opportunities of semantic analysis system, as a whole, essentially depend on successful construction of a middle part of the ontological structure. On the other hand, due to interaction of the middle level with upper one, text interpretation becomes more deeply and full: the explanatory resource **HiO** is used. If a new domain will be added, the middle level can be corrected.

The Lower Level of Ontology

The **HiO** lower level is produced to bind together text keywords to concepts of the middle level of ontology: here the concepts of the middle level can be determined by means of text words. On this level, two blocks of the interconnected concepts are stipulated.

One of them – *the interdisciplinary block* – is intended for text processing of the general-thematic discourses. Other – *the domain block* – is synthesized under concrete domain. These two blocks are connected mutually and correlate to concepts of middle level too. When on an input of ontology appears some text interdisciplinary block always works; the domain block becomes more active on the professional text.

The choice of concepts field and connections between them for the lower level makes by engineer on knowledge and domain expert together. There are technologies to process linguistic resources, for example, the software complex "RuThes" and other [1, 2, 4] which allow to synthesize domain ontology. However, if to be oriented on simpler problem – the localization of the document theme – much more simple procedures can be offered.

One of them consists in indexing: the words and terms of common and professional glossaries (which are represented in database) can be connected with concepts of lower level of ontology. The other way is proposed by T.Taran: some situation or scene is determined by concept lattices [15].

After construction the lower level of ontology, **HiO** synthesis comes to an end. However, received ontology represents only the theoretic-descriptive scheme, instead of the analysis system. For a text processing, it is necessary to connect ontology with a special dictionary of natural language, which is contained in ones memory – database.

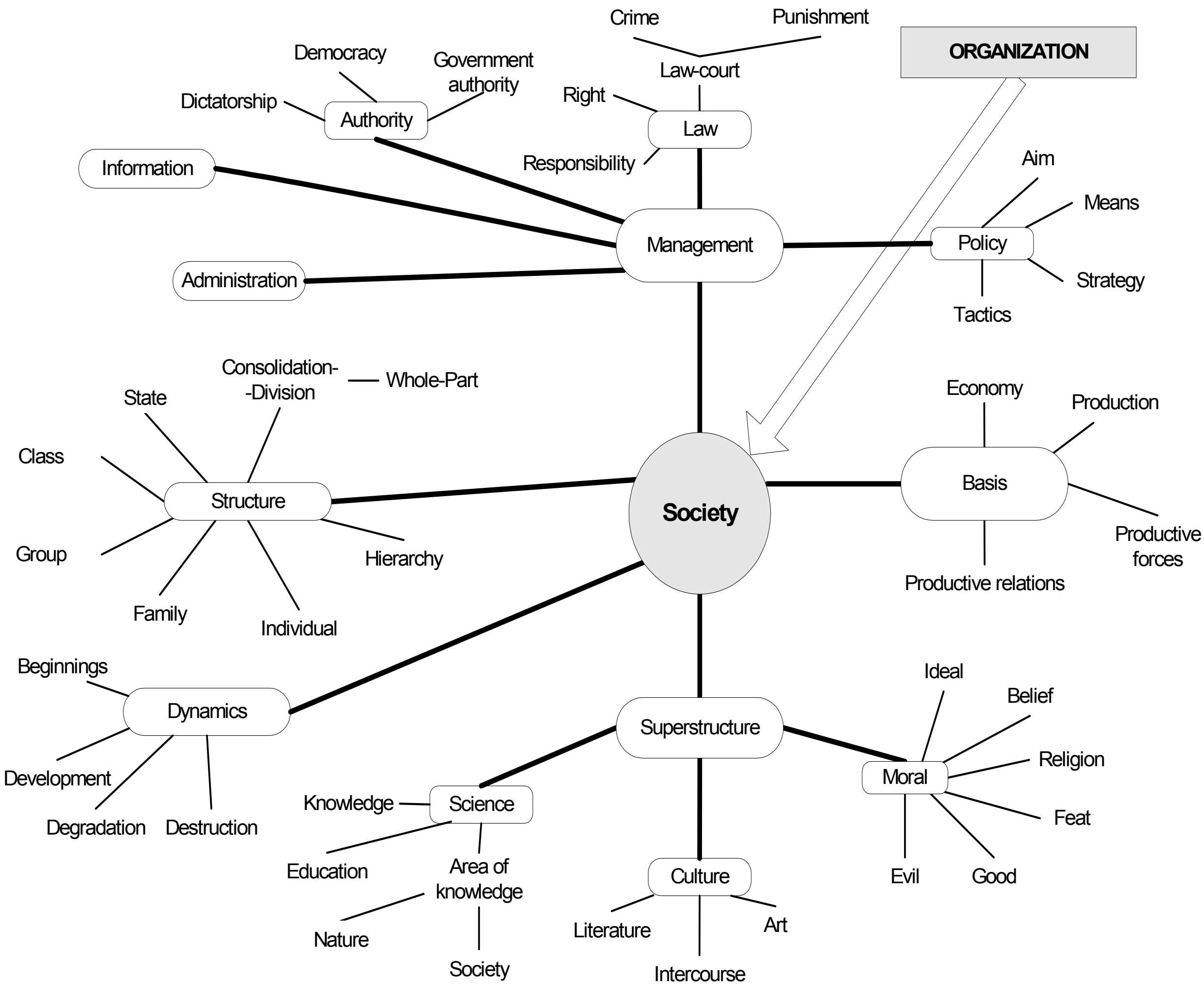


Fig. 3. A fragment of associations ontology structure of a middle level for a category "Organization".

Text-processing Procedure

The binding together of the text and the ontology is made through the dictionary of Russian. The dictionary reflects lexicon of a natural language. When the real NL text is analyzed, first of all the set of keywords is discover. These keywords should make active some elements of the lower level of ontology. Which elements will be made active, will specify the list of the indexes, appropriated to dictionary elements. This list in an obvious kind sets associative connections between the given elements and concepts of the lower level. Hence, the concrete text word through the dictionary stimulates a subset of concepts and connections of the lower level, and through these concepts the signal transfers to middle and upper levels of **HiO**. As a result, in all ontological network automatically some semantic trajectory of an entrance word is localized.

This trajectory is possible to use in the text-processing system, namely: a) for deeper interpretation of the text, or b) – as an initial material for repeated, purposeful disclosing a theme in enriched context.

Conclusion

As a result of researches is developed hierarchical structure of three-level associations ontology, which differs by the following:

– unites in uniform structure the general categories of the description of the world (on upper level) with the conceptual environment of interdisciplinary knowledge (on a middle level) and with the topical concepts at the lower level. Ontology supposes inclusion of the new blocks – models of domains – without alteration of upper level and with expansion of a middle level of a network;

– the network model of associations ontology is the simple and constructive scheme, which allows to trace in the text the theme that was given. Synthesis HiO practically excludes greater expenditures of labour on viewing of the texts collection, because a priori is based on known natural-scientific knowledge;

– the HiO serves as a construct with well-founded basis of scientific general categories. At a level metaontology the bases of categories dichotomy are precisely well founded and their conceptual completeness is proved. For a semantic description of environment, informal (associative) connections are widely used. The semantic trajectories of the conceptual analysis, received as a result the text-processing, help to interpret any theme in a context of universal human knowledge.

Based on hierarchical three-level associations ontology the new version of "Konspekt" system is developing.

Bibliography

1. Dobrov B.V., Loukachevitch N.V. Linguistic ontology of natural sciences and technologies for applications in sphere of information search. Scientific notes of the Kazan state university. Series Physical and mathematical sciences. – 2007. Volume 149, book 2. pp. 49-73. In Russian.

2. Loukachevitch N.V., Nevzorova O.A. Aviaontology-2004: the analysis of a modern state of a resource // Computer linguistics and intellectual technologies: Works of the International seminar Dialogue'2004 (June, 2-7nd 2004г.) // Edited by I.M.Kobozeva, A.S.Narinjani, V.P.Selegej. – M.: Nauka – 2004. – Vol.2 - pp. 424-430. In Russian.

3. Palagin O.V., Petrenko M.G. Construction of abstract model of language- ontological information system. // Mathematical machines and systems. - 2007. - №1. - pp.42-50. In Ukrainian.

4. Artemieva I.L. Multilevel ontologies for domains with complicated structures. // Proceedings of the XIII-th International Conference "Knowledge-Dialogue-Solution" – Varna, 2007 Volume 2 Sofia, Institute of Information Theories and Applications FOI ITHEA, Bulgaria– 2007 pp. 403-410

5. Morkovkin V.V. Ideographic dictionaries. M.: Mosc. st. uni-ty, 1970. – 71 p. In Russian.

6. Baranov O.S. The Ideographic dictionary of Russian, M. – 2002, – 1200 p. In Russian.

7. Miller G., Beckwith R., Fellbaum C., Gross D. and Miller K. Five papers on WordNet. - CSL Report 43. Cognitive Science Laboratory, Princeton University, 1990.

8. Mahesh, Kavi. Ontology development for machine translation: ideology and methodology. Memoranda in Computer and Cognitive Science MCCS–96–292, New Mexico State University. 1996.

9. SUO, (2001), The IEEE Standard Upper Ontology web site, http://suo.ieee.org.

10. Sowa, John F. Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, © 2000. - 594pp.

11. Guarino N. Formal Ontology and Information Systems. In N. Guarino (ed.) Formal Ontology and Information Systems. //Proceedings of FOIS'98. - Trento, Italy. - 1998. - 6-8 June. – IOS Press, Amsterdam. – pp.3-15.

12. Grishina O.V. Information search in Internet // Intellectual technologies and systems. Collected articles of post-graduate and students. Issue 2 // Edited by Yu.N.Filippovich. – M.: Publishing house MGUP, – 1999. – pp. 18-24. In Russian.

13. Gladun V.P., Velychko V. Yu., Svyatogor L.A. Thematic analysis of natural language texts. // Computer linguistics and intellectual technologies: Works of International Conference “Dialogue 2006” (Bekasovo, 31 May – 4 June, 2006) // Edited by A.S.Narinjani, V.P.Selegej. – M.: RSUH Publishing Centre. – 2006. pp. 115-118. In Russian.

14. Kaziev V.M. Introduction in the analysis, synthesis and modelling of systems. – INTUIT.RU, Binom. Laboratory of knowledge. – 2006. – 248p. In Russian.

15. Taran T.A., Zubov O.A. Artificial intelligence: the theory and applications. Lugansk. – 2006. – 240 p. In Russian.

Authors' Information

Victor Gladun, Vitalii Velychko, Leonid Svyatogor – *V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: aduis@rambler.ru*

UPGRADABLE TREE-LEVEL EDITOR OF METAONTOLOGIES, ONTOLOGIES AND KNOWLEDGE OF CHEMISTRY AND ITS DEVELOPMENT ON THE BASIS OF MULTILEVEL CHEMISTRY ONTOLOGY[1]

Irene Artemieva

***Abstract**: Development of upgradable tree-level editor of metaontologies, ontologies and knowledge for a chemistry intellectual system is described. A fragment of multilevel chemistry ontology is given. A dialogue scenario for editing ontologies of the second level is described. Data base schemes for representing ontologies and knowledge are defined. A way for adding graphical components to the editor is described.*

***Keywords**: Upgradable multilevel editor of ontologies and knowledge, domain ontology, chemistry ontology, domain knowledge*

***ACM Classification Keywords**: I 2.5 – Expert system tools and techniques.*

Introduction

While solving applied chemical tasks, researchers have to use ontologies and knowledge of different chemical domains and, in turn, solve nested tasks (as subtasks) of these domains. So the computer systems integrating these ontologies and different domains knowledge for solving chemical tasks are needed. As a scientific domain is being developed so the computer systems must be upgradable. In other words, from one hand it must allow user to add new ontologies and knowledge of new chemical domains, and from the other hand it must allow to add the new program components for solving applied tasks.

One of such kind of systems is the specialized computer knowledge bank for chemistry [Artemieva, Reshtanenko, 2006] – the expandable intellectual Internet-oriented program system for solving the diverse tasks from this professional domain, supporting the mechanisms for collective ontologies and data bases development and for adding new program components for solving applied tasks of this domain. To allow ontologies and knowledge bases to be expanded and developed, this knowledge bank is based on the multilevel chemical ontology [Artemieva et al, 2007], [Artemieva, 2008]. The upper level – called chemical metaontology – describes the structure of ontologies of different chemical domains, also known as the meta-ontologies of chemical domains. Each meta-ontology of chemical domain describes the structure of several representations of nested sub-domain ontologies. In its turn, sub-domain ontology describes the structure of information representation in the sub-domain knowledge base.

Chemist ordinary deals with the specialized objects as “compound structured formula”, “spectrum” and so on. The knowledge of such objects is represented in the traditional for chemistry graphical symbols. That’s why the knowledge editors must allow using specialized graphical editors, which may be called by ontology. For example, if some property of an object is the structured formula, than the structured formula editor for assigning this property must be called. The set of possible graphical object types may be expanded in the future; it requires new corresponding graphical editors. So the editor imbedded into the specialized chemical knowledge bank must be patchable with such components.

---

1 This paper was made according to the program № 2 of fundamental scientific research of the Presidium of the Russian Academy of Sciences, the project 09-I-П2-04

There are editors [Corcho et al, 2003], [Denny, 2004] allowing to create the domain ontologies by defining the concepts (classes) and their hierarchy. An ontology created is used for editing the domain knowledge. Knowledge elements are represented in these editors as the elements belonging to classes described in the ontology. Another approach to creation of the knowledge editors controlled by the metainformation (ontology of information) is described in the paper [Kleschev, Orlov, 2006]. Ontology is represented as the semantic net. Knowledge is another semantic net and its structure is defined by the ontology. But the methods of creating the specialized multilevel editors which allow addition of the special components for special objects (including graphical objects) editing are still not described in the literature.

The purpose of this paper is to describe the method of developing the expandable specialized 3-level editor for chemical metaontologies, ontologies and knowledge, based on multilevel chemical ontology [Artemieva et al, 2007].

The Structure of Multilevel Ontology of Chemistry

A structure of multilevel chemical ontology is represented on Fig. 1. The ontology of level 4 is a chemistry metaontology. The ontology of level 3 is a set of metaontologies of different chemical domains, i.e. it is an array of the modules corresponding to the chemical domains. Terms of the metaontology of a chemical domain are representatives of sets of terms of the metaontology of chemistry. The ontology of level 2 for each chemical domain is a set of ontologies of different sub-domains – modules of the ontology of level 2. Each module of the ontology of level 2 contains the definition of linked sets of terms. The ontology of level 1 for each chemical domain is a set of knowledge of sub-domains – modules of the ontology of level 1.

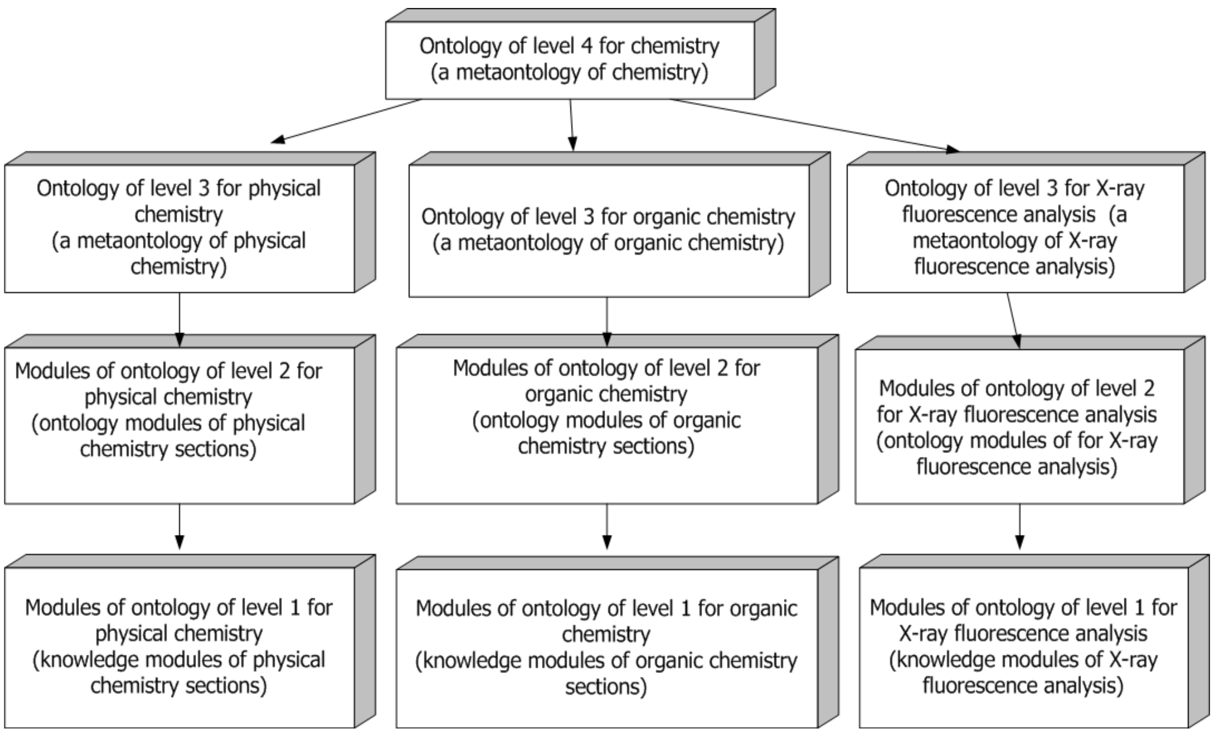


Fig. 1. The structure of multilevel ontology of chemistry

For example, in the ontology of level 2 for physical chemistry [Artemieva, Tsvetnikov, 2002] (Fig. 2) there are the following modules: “Properties of elements”, “Properties of substances”, “Properties of reactions”, “Introduction to thermodynamics”, “Thermodynamics. Chemical properties”, “Thermodynamics. Physical properties”, “Thermodynamics. Relation between physical properties and chemical properties”, “Chemical kinetics”. The first three modules define terms that describe properties of objects of a corresponding type. The module “Introduction to thermodynamics” defines terms used to describe general properties of thermodynamic systems and their components. Conditions of a thermodynamic system can change during a physicochemical process. Conditions of a process are assigned at discrete moments of observation. “Thermodynamics. Chemical properties” defines terms used to describe chemical changes of a substance during a process without taking into account phase changes. “Thermodynamics. Physical properties” define terms used to describe phase changes of a substance during a process without taking into account chemical changes. “Thermodynamics. Relation between physical properties and chemical properties” defines terms used to describe physicochemical processes. Finally, “Chemical kinetics” defines terms use to describe the dynamics of processes.

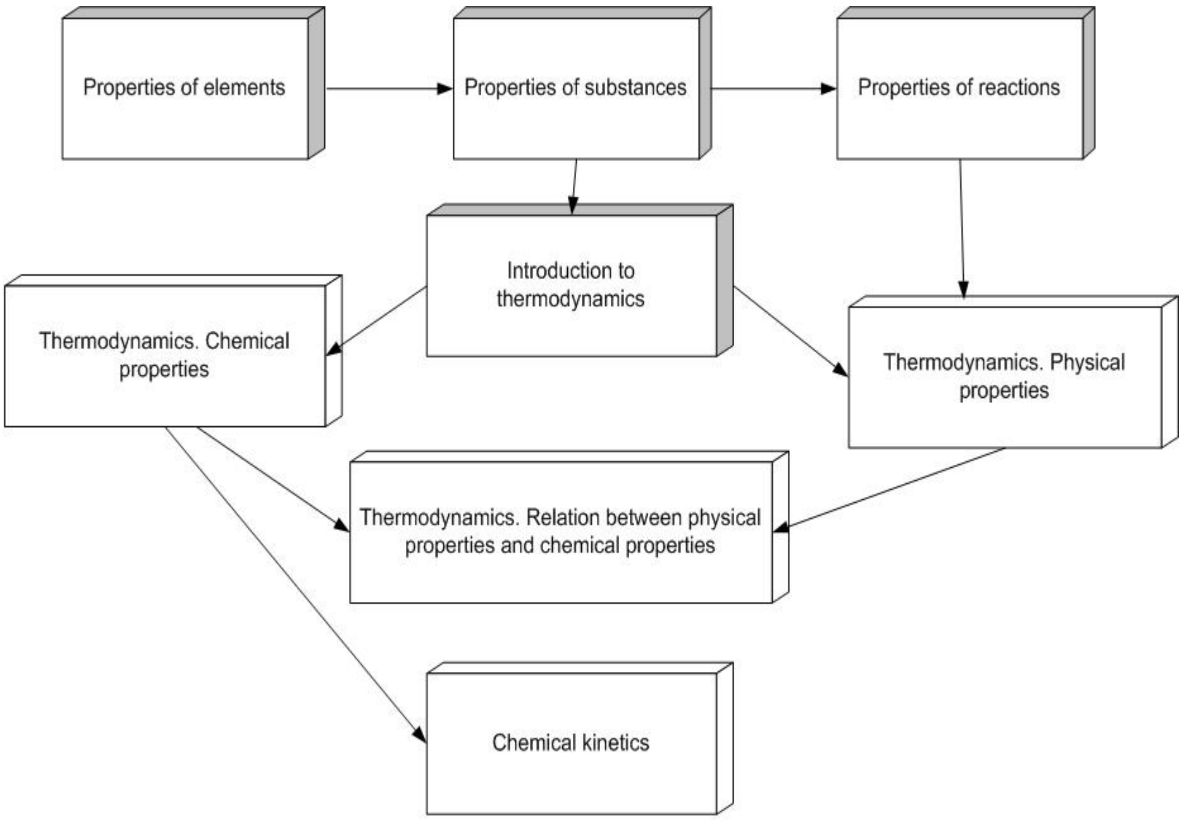


Fig. 2. The modules of the ontology of level 2 for physical chemistry

The ontology of level 2 for organic chemistry contains 26 modules (fig. 3) [Artemieva et al, 2005]]. It uses terms of the ontology of physical chemistry. This ontology defines terms for describing structural properties of compounds, molecular configuration, mechanisms of reactions, etc.

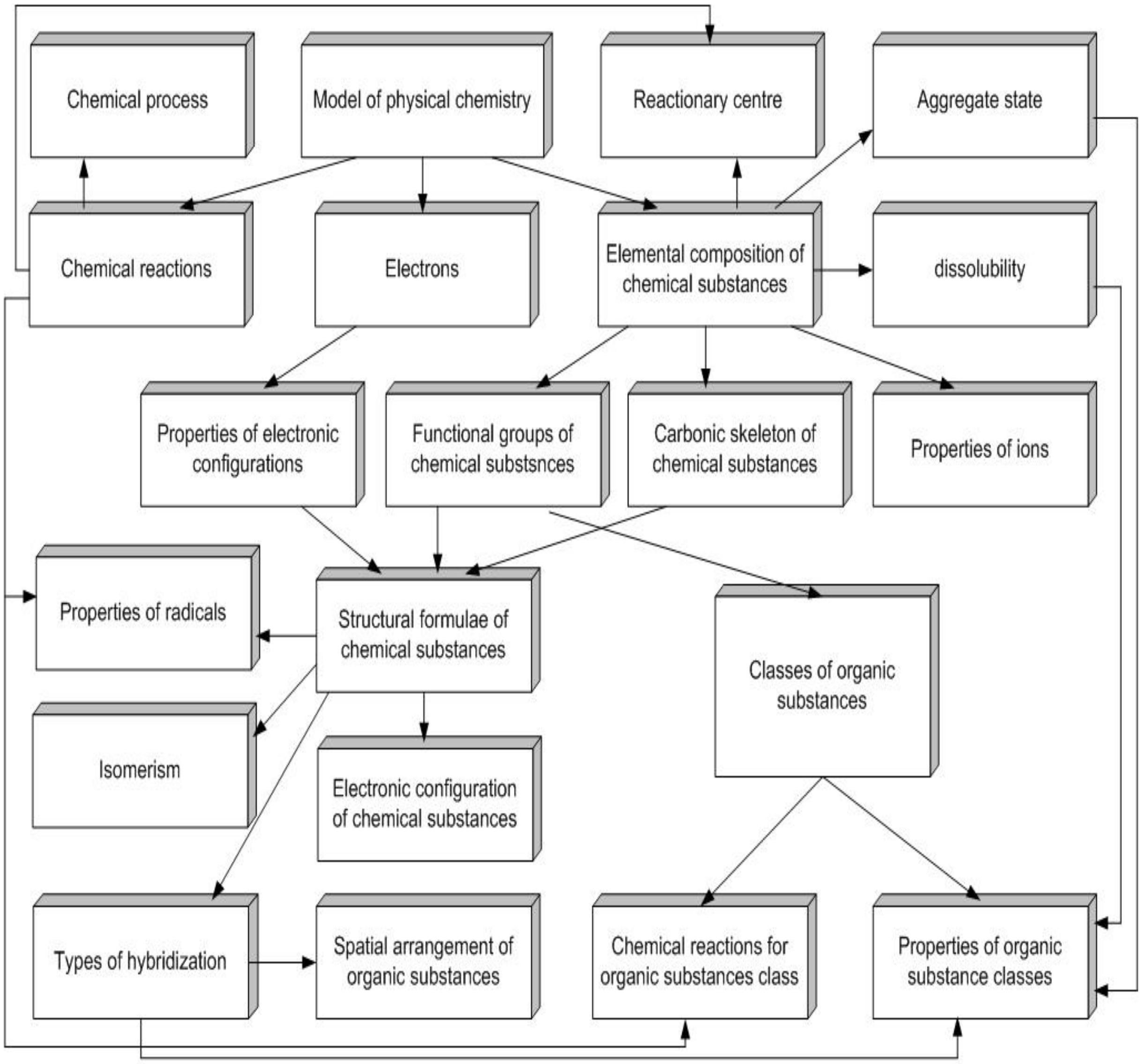


Fig. 3. The modules of the ontology of level 2 for organic chemistry

Components of Upgradeable Editor for Chemistry

Components of upgradeable tree levels editor of metaontologies, ontologies and knowledge of a chemistry are represented on the fig. 4.

Metaontologies of different chemical domains, ontology and knowledge modules of different sub-domains are information components of a specialized shell for a chemistry. Development and editing of information components is carried out by a tree level editor of metaontologies, ontologies and knowledge of a chemistry, the development of which is based on the chemical metaontology.

Tree level editor of metaontologies, ontologies and knowledge of a chemistry is to provide the development and editing of module metaontologies, ontologies and knowledge and to ensure the reusability of the modules for the development of ontologies and knowledge for new sections and subsections of the domain. Thus the development and editing of a module for the metaontology of a chemical section are to be controlled by the chemistry metaontology, while that of a knowledge module is to be controlled by the ontology of a section.

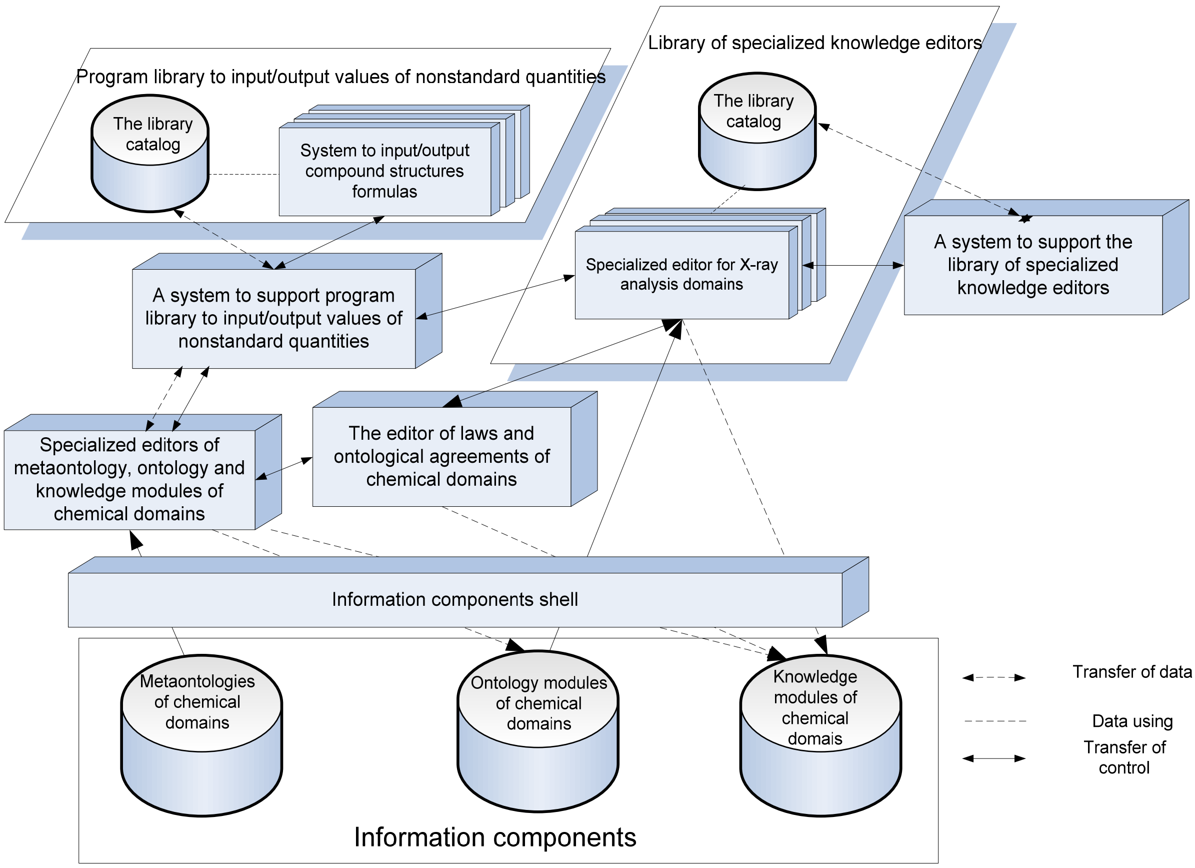


Fig. 4. Components of upgradeable editor for chemistry

Tree level editor of metaontologies, ontologies and knowledge of a chemistry provides the opportunity to choose that of the existing modules of the metaontology of a section that controls the editing of a module of ontology for this section under development. And also the editor provides the opportunity to choose that of the existing modules of the ontology of a section that controls the editing of a module of knowledge for this section under development.

Editors of ontology and knowledge are to provide the opportunity to specify the structured and not structured parts of the ontology and the structured and not structured parts of knowledge, i.e. a program component for these editors would be a specialized statement editor that allows to input the ontological agreements and laws of the domain.

Knowledge editor is to provide the input/output values of nonstandard quantities. An example of such quantities for chemistry is compound structured formula [Artemieva et al, 2006] there can be graphically represented. Therefore, while developing and editing knowledge, knowledge editor is to provide the opportunity to use the method of graphic representation of the values of nonstandard quantities accepted in the domain. The quantity, to which corresponds a value of a certain property, is specified by the ontology of level 2. Therefore knowledge editor is to provide an automatic choice (controlled by the ontology of level 2) of the means for graphic representation of values of nonstandard quantities.

Ontology editor interprets the ontology of level i while developing a module of the ontology of level i-1. Knowledge editor interprets the ontology of level 2 while developing knowledge module. The same ontology can be interpreted differently by different knowledge editors. Knowledge editors can vary by the ways of knowledge interpretation and by interface as well. It is obvious that for an editor aimed to interpret one ontology, not the whole class of ontologies, more easy-to-use interface and more comprehensible way of interpretation for an expert can be developed. Therefore three level editor for the chemistry should allow us the application of editors supporting different ways of interpretation of a module of the ontology of level 2, as well as providing an expert with the opportunity to choose the editor needed.

Thus, three level editor for the chemistry contains (Fig.4):

- specialized editor of metaontology and ontology for different chemistry domains,
- knowledge editor controlled by any ontology of level 2,
- program library for input/output values on nonstandard quantities,
- systems to input/output values of nonstandard quantities,
- a library of specialized knowledge editors each of which is controlled by a specialized ontology of level 2,
- statement editor allowing to specify laws and ontological agreements of the domain.

A Class of Models for Domains with Complicated Structures

Unenriched system of logical relationship O^m *of the level m* (where $m \geq 2$) *is three-tuple* $<\Phi^m, P^m, C^m>$, where Φ^m is a set which consists of notion definitions of the level m, constraints on value sets of the notions, and also relations between notion values; P^m is a set of parameters of the level m, and C^m is a set of constructor definitions of the level m. The sets P^m and C^m can be empty. If m=2 then $C^2 = \varnothing$.

When domain with complicated structure is modeled, then every unenriched system of logical relationship O^m is a module of the model of ontology of the level m for the domain.

Let us define a form of the sentences from the set Φ^m.

1) sort n: M is definition of a basic notion where n is the notion name and M is the notion denotation (extensional interpretation of the notion); M defines possible set of values for the notion with name n.

If denotation of notion with name n is defined as a mapping $(M_1 \times M_2 \times \ldots \times M_n \rightarrow M_0)$ where $M_1, \ldots, M_n, M_0$ are sets and $M_0 \neq$ {true, false} then n is a functional name. If denotation of notion with name n is defined as a mapping $(M_1 \times M_2 \times \ldots \times M_n \rightarrow \{true, false\})$ then n is predicate name. In other cases n is subject name.

2) $(v_1: M_1)$ $(v_2: M_2(v_1))$ … $(v_q: M_q(v_1, v_2,\ldots, v_{q-1}))$ $(n: M(v_1, v_2,\ldots, v_q))$ sort n: $M_0(v_1, v_2,\ldots, v_q)$ is definition of a set of basic notions names of which are hidden by parameters $v_1, v_2,\ldots, v_q$. Here $v_1, v_2,\ldots, v_q$ are variables, M_1 is a value set for the variable v_1, $M_2(v_1)$ is a value set of the variable v_2 depending on the value of $v_1,\ldots, M_q(v_1, v_2,\ldots, v_{q-1})$ is a value set for the variable v_q depending on values of $v_1, v_2,\ldots, v_{q-1}$; $M(v_1, v_2,\ldots, v_q)$ is definition of name set for basic notions by using parameters-variables $v_1, v_2,\ldots, v_q$; n is a variable, $M_0(v_1, v_2,\ldots, v_q)$ is a formula to determine denotations of all the notions from the set $M(v_1, v_2,\ldots, v_q)$) by using values of variables $v_1,v_2,\ldots,v_q$.

Parameters can hide functional, predicate or subject names.

3) $n \equiv t$ is definition of auxiliary notion where n is the notion name and t is the notion value or a formula to determine the notion value by using values of other notions.

We will distinguish functional, predicate and subject names of auxiliary notions.

If the value of a notion with n name is specified with λ-term the value of which is logical, n is a predicate name of an auxiliary notion. If the value of λ-term is not logical, n is a functional name of an auxiliary notion. A notion with n name is a subject name of an auxiliary notion if λ-term is not used when specifying.

4) (v_1: M_1) (v_2: $M_2(v_1)$) … (v_q: $M_q(v_1, v_2,…, v_{q-1})$) (n: $M(v_1, v_2,…, v_q)$), n ≡ $t(v_1, v_2,…, v_q)$ is definition of a set of auxiliary notions names of which are hidden by parameters $v_1, v_2,…, v_q$. Here $v_1, v_2,…, v_q$ are variables, M_1 is a value set for the variable v_1, $M_2(v_1)$ is a value set of the variable v_2 depending on the value of $v_1,…$, $M_q(v_1, v_2,…, v_{q-1})$ is a value set for the variable v_q depending on values of $v_1, v_2,…, v_{q-1}$; $M(v_1, v_2,…, v_q)$ is definition of a name set for auxiliary notions by using parameters-variables $v_1, v_2,…, v_q$; n is a variable, $t(v_1, v_2,…, v_q)$ is a formula to determine denotations of auxiliary notions (values of n names from $M(v_1, v_2,…, v_q)$) by using values of variables $v_1,v_2,…,v_q$.

Parameters can hide functional, predicate and subject names of auxiliary notions.

5) (v_1: M_1) (v_2: $M_2(v_1)$) … (v_q: $M_q(v_1, v_2,…, v_{q-1})$) $f(v_1, v_2,…, v_q)$ or f is a specification of constraints on value sets of the notions and relations between notion values. Here $v_1, v_2,…, v_q$ are variables, M_1 is a value set for the variable v_1, $M_2(v_1)$ is a value set for the variable v_2 depending on the value of $v_1,…$, $M_q(v_1, v_2,…, v_{q-1})$ a value set for the variable v_q depending on values of $v_1, v_2,…, v_{q-1}$; $f(v_1, v_2,…, v_q)$ is a formula with $v_1, v_2,…, v_q$, f is a formula without any variables.

In sentences of modes (2), (4) and (5) the sequence (v_1: M_1) (v_2: $M_2(v_1)$) … (v_q: $M_q(v_1, v_2,…, v_{q-1})$) is called prefix.

Sentences of modes (1) and (2) define basic notions of the ontology of a section or subsection of a domain with complicated structure. Subject names specify names of sets of objects of this section. Functional and predicate names specify names of relations among the objects of the section.

Let us consider the examples of sentences of modes (1) and (2) written with means of applied logic language class defined in [Kleshchev, Artemieva, 2005].

The following sentence of mode (1): sort Types of objects: {}N \ ∅ defines the basic notion with "Types of objects" name with the volume of which is a set of all subsets of the denotation set excluding an empty one.

The following sentence of mode (1): sort Types of object components: (Types of objects → {}Types of objects) defines the basic notion with "Types of object components" (functional) name the volume of which is a set of functions the definition area of which is "Types of objects" set and the value area is a set of all possible subsets of "Types of objects" set.

The following sentence of mode (1): sort Phase equilibrium: I[1, Number of process steps] → {true, false} defines the basic notion with “Phase equilibrium” (predicate) name the volume of which is a set of predicates the definition area of which is a set of integers at least equal to 1 and less than or equal to the value assigned by "Number of process steps" term.

The following sentence of mode (2) defines a set of the basic notions the names of which are hidden by "Types of objects" parameter [Artemieva, 2008]: (object type: Types of objects) sort object type: {}(R∪N∪I∪L). Each type of object is a set of objects; each object can be named, represented with a number, can be logical value.

Sentences of modes (3) and (4) are used to specify auxiliary notions of a section or subsection of a domain. Let us consider the examples of sentences of mode (3):

1. intensive parameters ≡ {temperature, pressure, density, chemical potential}

defines a basic notion of the section with “Intensive parameters” name of physical chemistry the value of which is a set containing the following elements: “temperature”, “pressure”, “density”, “chemical potential”.

2. possible formula of substance ≡ (∪(n: I[1, ∞)) {(v: ((× chemical elements, I[1,∞))⇑ n)) (&(i: I[1, length(v)]) (&(j: I[1, length(v)] \ {i}) π(i, v) ≠ π(j, v)))}) ∪ (∪(n: I[1, ∞)) {(v: ((× chemical elements ∪ possible formula of substance, I[1, ∞))⇑ n)) (&(i: I[1, length(v)]) (&(j: I[1, length(v)] \ {i}) π(i, v) ≠ π(j, v)))}) defines a basic notion with "Possible formula of substance" name the value of which is a set of all possible sequences (simple or complex) of components of a formula of a chemical compound. Each simple component is a pair consisting of a chemical element and its index; each complex component is a pair the first element of which is a component of the formula, the second element is its index.

3. index ≡ (λ (v1: possible formula of substance) (v2: {(v3: chemical elements ∪ possible formula of substance) belongs to compound (v1, v3)}) π(2, component(v1, component number(v1, v2)))) defines an auxiliary notion with “Index” name the formula to determine denotation of which is specified with λ-term.

When modeling domains with complicated structures, sentences of mode (3) are used to determine a set of values belonging to some nonstandard quantity. The left part of such sentences defines the name of a nonstandard quantity (subject name), and the right part defines the way of constructing elements of the value set of this quantity. The above example 2 defines a nonstandard quantity with “Possible formula of substance”, the right part of the sentence specifies the rule for constructing formulas. Sentences of mode (3) are also used to define operations and relations with elements of nonstandard quantities. The left part of such sentences specifies functional (a sign of the operation) or predicate (a sign of relation) name, and the right part is λ-term. The above example 3 defines “Index” operation for the value with “Possible formula of substance" name.

Constraints on value sets of the notions and relations between notion values are specified by sentences of mode (5). For instance, sentence Type1: Types of entities) (Type2: Types of entities \ {Type}) j(Type1) ∩ j(Type2) = ∅ specifies relations between values of notions names of which are hidden by “Types of entities” parameter: sets of entities of different types do not intersect.

A lot of sentences of mode (5) which contain only the names of Pm set represent the integrity constraints for m-1 ontologies. The rest sentences of mode (5) specify some integrity constraints for ontologies of the following levels, some integrity constraints for domain knowledge, and some relations between domain knowledge and reality.

Elements of the set of P^m parameters are a subset of the set of names of basic notions of m level defined in sentences of mode (2) included in Φ^m. All P^m elements specify terms to describe m-1 ontologiy. If P^m set ≠ ∅, O^m is unenriched system of logical relationship of the level m with parameters. Otherwise, O^m unenriched system of logical relationship of the level m without parameters.

When m≥3, subject parameters are names of sets of terms m-1ontology, functional and predicate parameters are names of functional and nonfunctional relations between terms of m-1 ontology. When m=2, subject parameters are names of sets of objects of domain section, functional and predicate parameters are names of functional and nonfunctional relations between objects.

"Types of objects" is an example of subject parameter of level 4 ontology for chemistry. Its value specifies many object types of chemistry section – terms of metaontology of chemistry section. For example, for physical chemistry, there are such types as "Chemical elements", "Chemical substances", “Chemical reactions."

“Types of object components" is an example of functional parameter of level 4 ontology for chemistry. Its value specifies relation "Components of object having type t_1 are objects having types $t_2,\dots,t_k$" where t_1 ranges over values of "Types of objects" parameter.

Elements of set $C^m = \{c^m_1, c^m_2,\dots, c^m_z\}$ are constructors defining a way of constructing sorts of terms of ontologies of levels less than m. Each c^m_i relates the name of the i^{th} constructor to λ - term:

t ≡ (λ(v11: M11) (v12: M12) … (v1q1: M1q1) (λ(v21: M21(v1,…vq1)) (v22: M22(v1,…vq1)) … (v2q2: M2q2(v1,…vq1)) … (λ(v^s_1: $M^s_1(v_1,…v_{qs-1})$) (v^s_2: $M^s_2(v_1,…v_{qs-1})$) … (v^s_{qs}: $M^s_{qs}(v_1,…v_{qs-1})$) $M(v^1_1,…v^s_{qs})$)…)), where

t is constructor name; (λ(v^1_1: M^1_1) (v^1_2: M^1_2) … (v^1_{q1}: M^1_{q1}) (λ(v^2_1: $M^2_1(v_1,…v_{q1})$) (v^2_2: $M^2_2(v_1,…v_{q1})$) … (v^2_{q2}: $M^2_{q2}(v_1,…v_{q1})$) … (λ(v^s_1: $M^s_1(v_1,…v_{qs-1})$) (v^s_2: $M^s_2(v_1,…v_{qs-1})$) … (v^s_{qs}: $M^s_{qs}(v_1,…v_{qs-1})$) $M(v^1_1,…v^s_{qs})$)…)) is λ-term; s is constructor order, $1 \le s \le m-1$; v^1_1, v^1_2,…, v^s_{qs} are constructor parameter; parameters v^{i+1}_1, v^{i+1}_2,…, v^{i+1}_{qi+1} are parameters of (s-i)th order; M^1_1, M^1_2,… , M^1_{q1},…, M^s_{qs} are value sets of constructor parameters; definition of at least one of sets M^1_1, M^1_2,… , M^1_{q1} depends on parameters of the level m; $M(v^1_1,…v^s_{qs})$ is definition of set depending on parameters $v^1_1,…v^s_{qs}$.

Let us consider the examples of defining constructors of the following sorts:

1. Own properties of chemical process ≡ (λ (Area of possible values: {}(Value sets ∪ {}Value corteges)) (I[1, Number of process step] → Area of possible values))

Are defined by the constructor of the first order the parameter of which is "Area of possible values". When the parameter value is specified, the constructor makes a sort as a set of functions, the arguments of each are number of process step, and the result is an element of the set specified by the parameter value.

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

Are defined by the constructor of the second order. "Object type" is the parameter of the second order, and "Area of possible values " is the parameter of the first order. When the value of "Object type" parameter is specified, the constructor makes a set of functions, the definition area of each is a set of values or a set of m corteges; value area is a set of functions the argument of each is entity of t-type, and the result is an element of m set.

Enrichment of Unenriched System of Logical Relationship

Transition from the ontology model of the level m to the ontology model (when m≥3) or knowledge model (when m=2) is achieved by specifying enrichment of O^m system. At this stage, enrichment components and rules of construction of O^{m-1} are defined, if there is some enrichment of O^m system specified.

Specifying the level m-1 ontology in terms of the level m ontology involves definition of:

- notions of the level m-1 ontology the names of which are hidden by the level m ontology parameters;
- notions of the level m-1 ontology using sort constructors;
- ontological agreements of the level m-1 ontology;
- sort constructors of the level m-1;
- the level m-1 parameters.

Therefore *enrichment of unenriched system of logical relationship* O^m is a set of $k^m = <\alpha^m_P, EN^m, ER^m, EC^m, EP^m>$, where α^m_P is interpretation of the level m parameters, EN^m is a set of definitions of the level m-1 notions, ER^m is a set of relations between the level m-1 notions, EC^m is a set of definitions of the level m-1 constructors, EP^m is a set of the level m-1 parameters. Sets EN^m, ER^m, EP^m, EC^m can be empty but $EN^m \cup ER^m \cup EP^m \cup EC^m \neq \emptyset$.

When m ≥ 3, α^m_P interpretation defines terms of the level (m-1)th ontology, constraints on their value sets and relations between the term values: the set of ontology terms make values of subject parameters, constraints on their value sets and relations between the term values are specified by the interpretation of functional and predicate parameters.

For instance, enrichment of the level 4 ontology specifies the following terms of the level 3 ontology (value of parameter "Types of objects") for organic chemistry: Chemical elements, Chemical substances, Chemical reactions, Inorganic substances, Organic substances, Functional groups, Radicals, Reaction stimulators, and determines (for each object type) objects of what types are its components (value of parameter "Type of object components").

When m = 2, α^2_P interpretation relates objects or sets of objects of the defined section of a domain to object parameters, and functional and non-functional relations between objects – to functional and predicate parameters.

For example, value of parameter "Chemical elements" of the level 2 ontology defines a set of names of these elements in the module of knowledge of physical chemistry, and value of parameter Reaction stimulators defines the relation "to be stimulators" between reaction and a set of chemical compounds.

Elements of EN^m define terms of the ontology section, specifying sets of their values by using one of the constructors of the first order of C^m. Each definition of the level m-1 notion (element of EN^m) relates the set which is the result of applying the constructor of the first order $t \equiv (\lambda(v_1: M_1)\ (v_2: M_2) \dots (v_q: M_q)\ M(v_1,\dots v_q))$ defined at the level m to the notion name:

sort p: $t(c_1, c_2,\dots c_{q1})$, where p is the notion name, $t(c_1, c_2,\dots c_{q1})$ is application of the constructor of the first order defined at the level m, where $c_1 \in J_{\alpha,\theta1}(M_1)$, $c_2 \in J_{\alpha,\theta2}(M_2)$, …, $c_q \in J_{\alpha,\theta q}(M_q)$, where $\alpha = \alpha^m_P$, θ_1 is empty substitution $\theta_2 =\{v_1/c_1\},\dots,\theta_q=\{v_1/c_1,\dots,v_{q-1}/c_{q-1}\}$.

When the interpretation function α is specified, the application of the constructor of the first order t to the set of values $c_1, c_2,\dots c_q$ of parameters $v_1, v_2,\dots v_q$ gives the set $J_{\alpha,\theta}(M(v_1,\dots v_q))$, где $\theta = \{v_1/c_1, v_2/c_2,\dots v_q/c_q\}$.

For example, the module "Electrons" of organic chemistry contains the following definitions of terms using sort constructors:

- - sort the number of electrons of element: Own properties of elements(I[1, maximum number of electrons]);
- - sort electrons of element: properties of electronic levels of element({}(× main quantum number, azimuthal quantum number, magnetic quantum number, spin)).

Elements of ER^m, when m>2, specify ontological agreements of the section ontology, and when m = 2 – unstructured knowledge of the section. If ontologies of all sections can be represented verbally, ER^m is empty. When writing relations between the level m-1 notions (elements of ER^m), names of basic and auxiliary notions defined in EN^m and Φ^m and not belonging to P^m are used.

Elements of EC^m specify names of constructors of the (m-1)th level. Each constructor definition (element of σ^m) matches the name of the constructor with λ - term which is the result of the constructor application $t \equiv (\lambda(v^1_1: M^1_1)\ (v^1_2: M^1_2) \dots (v^1_{q1}: M^1_{q1})\ (\lambda(v^2_1: M^2_1(v_1,\dots v_{q1}))\ (v^2_2: M^2_2(v_1,\dots v_{q1})) \dots (v^2_{q2}: M^2_{q2}(v_1,\dots v_{q1})) \dots (\lambda(v^s_1: M^s_1(v_1,\dots v_{qs-1}))\ (v^s_2: M^s_2(v_1,\dots v_{qs-1})) \dots (v^s_{qs}: M^s_{qs}(v_1,\dots v_{qs-1}))\ M(v^1_1,\dots v^s_{qs}))\dots))$ defined at the level m m: $t1 \equiv t(c_1, c_2,\dots c_{q1})$, where

- - t_1 is constructor name;
- - $c_1 \in J_{\alpha,\theta1}(M^1_1)$, $c_2 \in J_{\alpha,\theta2}(M^1_2)$, …, $c_{q1} \in J_{\alpha,\theta q1}(M^1_{q1})$, где $\alpha = \alpha^m_P$, θ_1 is empty subsitution, $\theta_2 =\{v_1/c_1\},\dots,\theta_{q1}=\{v_1/c_1,\dots,v_{q1-1}/c_{q1-1}\}$;

- - $t(c_1, c_2,\ldots c_{q1})$ is application of the level m constructor t, where the constructor order s satisfies the condition: $1 \le s \le m-1$.

When the interpretation function α is specified, the application of the constructor of the s^{th} order t to the set of values $c_1, c_2,\ldots c_{q1}$ of parameters $v_1, v_2,\ldots v_{q1}$ gives the constructor of the $(s-1)^{th}$ order $t(c_1, c_2,\ldots c_{q1})$ the value of which is λ - term $(\lambda(v^2_1: J_{\alpha,\theta}(M^2_1))\ (v^2_2: J_{\alpha,\theta}(M^2_2))\ \ldots\ (v^2_{q2}: J_{\alpha,\theta}(M^2_{q2}))\ \ldots\ (\lambda(v^s_1: J_{\alpha,\theta}(M^s_1))\ (v^s_2: J_{\alpha,\theta}(M^s_2))\ \ldots\ (v^s_{qs}: J_{\alpha,\theta}(M^s_{qs}))\ J_{\alpha,\theta}(M(v^1_1,\ldots v^s_{qs})))\ldots)$, где $\theta=\{v_1/c_1, v_2/c_2,\ldots v_{q1}/c_{q1}\}$.

For example, in the level 3 ontology for organic chemistry there are the following sort constructors:

- Own properties of elements $\equiv$ Own properties of entities(Chemical elements)
- Own properties of substances $\equiv$ Own properties of entities(Chemical substances)
- Own properties of organic compounds $\equiv$ Own properties of entities(Organic compounds)

EPm is a subset of the set of names of basic notions defined in ENm and Φm and not belonging to Pm. Elements of EP^m are terms for describing the level m-2 ontology.

Enrichment k^m is possible for unenriched system of logical relationship O^m if the following conditions are met:

1. if $\alpha^m{}_P$ is possible interpretation function of parameters from set P^m, there is model α for T^m, where $\alpha^m{}_P$ is its restriction to a number of parameters;

2. if m > 2 and $P^m \ne \varnothing$, $\alpha^m{}_P \ne \varnothing$;

3. if m > 2 и $P^m = \varnothing$, $\alpha^m{}_P = \varnothing$;

4. if m=2, $\alpha^2{}_P \cup ER^2 \ne \varnothing$, $EC^2 = \varnothing$, $EP^2 = \varnothing$;

5. $EN^m \cup EC^m \ne \varnothing$;

6. there is at least one model for a set of sentences $\Phi^m \cup C^m \cup EN^m \cup \alpha^m{}_P \cup ER^m \cup EC^m$.

Let us define *the operation $O^m \bullet k^m$ of the application of enrichment* $k^m = <\alpha^m{}_P, EN^m, ER^m, EC^m, EP^m>$ to unenriched system of logical relationship $O^m =<\Phi^m, P^m, C^m>$. As a result of this operation, there is unenriched system of logical relationship of the $(m-1)^{th}$ level $O^{m-1} = <\Phi^{m-1}, P^{m-1}, C^{m-1}>$ with the following properties:

- $\Phi^{m-1} = \Phi^m \cup EN^m \cup \alpha^m{}_P \cup ER^m$;

- $P^{m-1} = EP^m$;

- $C^{m-1} = EC^m \cup C^m$.

Operation $O^m \bullet k^m$ determines how the system O^{m-1} is designed, using the enrichment k^m.

Let us denote $En_\alpha(O^m)$ – the set of possible functions of interpretation of parameters. The definition of possible interpretation function makes it clear that unenriched system of logical relationship O^m defines $En_\alpha(O^m)$.

The infinite set of all possible enrichments of Om is $En(O^m)$. Unenriched system O^m defines a set of unenriched systems of logical relationship $\{O^m \bullet k^m \mid k^m \in En(O^m)\}$ of the level (m-1). This means that any model of the ontology of the level m-1 is an element of $\{O^m \bullet k^m \mid k^m \in En(O^m)\}$.

Development of Editors of Information Components

The model of the level m ontology is the basis for developing the multilevel editor. The creation of each of the level m-1 ontologies of is controlled by this ontology. Each level i ontology controls the creation of the level i-1 ontology. The level 2 ontology controls the creation of the domain knowledge base module.

Designing the level i-1 ontology model includes specifying the enrichment of the level i-1 ontology model according to the definition of its structure in Chapter 2. Making the knowledge base module requires specifying the enrichment of the level 2 ontology model. Thus, the multilevel editor solves the task of creation of some information under control of the metainformation and implements the application of the enrichment [] to the metainformation model.

Let us consider the process of development of multilevel editor without using tool systems. Let the level m ontology and its model be developed and represented in the class of systems of logical relationship defined in this paper. Let us specify the partial order relation on the basis of the relationship between term values in terms of this ontology.

We will say that the value of the subject term t2 depends on the value of the subject term t1 if the definition of the term t2 sort uses the term t1 or there is ontological agreement in which the value of the term t2 is defined as a subset of values of the term t1.

We will say that the value of the functional term t2 depends on the value of the subject term t1 if the definition of the term t2 sort uses the term t1 or there is ontological agreement in which the value set of function denoted by the term t2 is defined as a subset of values of the term t1.

We will say that the value of the functional term t2 depends on the functional term t1 if the definition of the term t2 sort uses the term t1 or there is ontological agreement in which the value set of function denoted by the term t2 is defined as a subset of values of function denoted by the term t1.

We will say that the value of the predicate term t2 depends on the value of the term t1 if the definition of the term t2 sort uses the term t1.

We will say that the value of the term t2 specifying the name of the constructor depends on the value of the term t1 if the constructor definition uses the term t1.

In accordance with the partial order relation let us make a graph of relation between the names of parameters and level m ontology constructors. The upper level of the graph includes terms the values of which do not depend on other terms of the ontology. At the next level, there are terms the values of which depend on the upper level terms.

Let us use the relation graph designed for the level m ontology in the development of editor dialogue scenario in when forming the level m-1 ontology. Obviously, in accordance with the scenario it will always start with the input values of parameters of the level m ontology the names of which do not depend on the values of other parameters. Forming the ontology of each next level will also begin with entering parameter values of this level which do not depend on the values of other parameters.

The partial order relation between the terms of ontologies of all levels can be defined on the basis of the partial order relation between the terms of the level m ontology.

As the user specifies model enrichment of this ontology which controls the process of editing with the editor, its steps are defined by rules of enrichment specifying [].

The performance of the level i ontology editor under control of the level i+1 ontology starts with defining the value of a parameter of the ontology. In the ontology model, the parameter description (the name of the set of terms of the level i ontology) is as follows: sort M: {}N \ ∅. Knowledge engineer specifies the elements of the set M. The level i +1 ontology model also contains, in this case, sentences of one of the following types:

1. (v: M) sort v: M_1
2. (v: M) sort v: {} M_1
3. (v: M) sort v: $M_1 \cup M_2 \cup \ldots \cup M_n$
4. (v: M) sort v: {} $(M_1 \cup M_2 \cup \ldots \cup M_n)$

which specify sort (value set) for each term of the set M. Sets $M_1 \cup M_2 \cup \ldots \cup M_n$ included in the sort definition are specified in the level i+1 ontology. Let knowledge engineer define M = $\{t_1,t_2,\ldots,t_k\}$. The ontology editor relates sort to each term t_j $(1 \le j \le k)$ in accordance with its specification in the leveli+1 ontology model.

If in the level i+1 ontology model the term sort from the set M is specified by a sentence of type 1 (type 2), the ontology editor relates sort M_1 (sort {} M_1) to each term t_j ($1 \le j \le k$). If in the level i+1 ontology model the term sort from the set M is specified by a sentence of type 3 (type 4), the ontology editor offers the knowledge engineer to choose one of the sets $M_1, M_2, \ldots, M_n$.

Once the knowledge engineer has defined all terms of the formed ontology, they specify which of them can be parameters, i.e. will be used when designing the ontology of the next level.

If the level i+1 ontology model contains the constructor $t \equiv (\lambda(v^1{}_1: M^1{}_1)\ (v^1{}_2: M^1{}_2)\ \ldots\ (v^1{}_{q1}: M^1{}_{q1})\ (\lambda(v^2{}_1: M^2{}_1(v_1,\ldots v_{q1}))\ (v^2{}_2: M^2{}_2(v_1,\ldots v_{q1}))\ \ldots\ (v^2{}_{q2}: M^2{}_{q2}(v_1,\ldots v_{q1}))\ \ldots\ (\lambda(v^s{}_1: M^s{}_1(v_1,\ldots v_{qs-1}))\ (v^s{}_2: M^s{}_2(v_1,\ldots v_{qs-1}))\ \ldots\ (v^s{}_{qs}: M^s{}_{qs}(v_1,\ldots v_{qs-1}))\ M(v^1{}_1,\ldots v^s{}_{qs}))\ldots))$, where $M^1{}_1, M^1{}_2, \ldots, M^1{}_{q1}$ are the names of the level i+1 parameters, the ontology editor performs as a user wizard: the knowledge engineer is offered to specify the name of the defined term the sort of which is the result of application of λ-term to the cortege (this operation is defined in Chapter 2) made of the specified elements of sets $M^1{}_1, M^1{}_2, \ldots, M^1{}_{q1}$. The number of step restarts equals the strength of Cartesian product defined at the previous steps of sets $M^1{}_1, M^1{}_2, \ldots, M^1{}_{q1}$. All the terms specified at this step are constructors used to edit the level i-1 ontology.

The ontology editing consists of adding new terms and removing the existing ones. When you remove a term, the ontology editor requires confirmation of the removal and gives the knowledge engineer a follow-up if the term is used in the ontologies of the level i-1 based on the edited ontology of the level i. Knowledge editing means defining values of terms of the knowledge ontology. New terms are not specified. At the last step, editors of ontologies and knowledge call editor of ontological agreements and domain laws. Once the module of ontology or knowledge is fully developed, the implementation of ontological agreements specified by the ontology of a higher level on the bases of which the module of ontology or knowledge was created can be checked.

Database Structure for Storing Ontology and Knowledge

Second level ontology is stored by means of database control system. Each domain corresponds to its particular database with the same name. The structure of 2-nd level ontology is fixed by the 3-d level ontology [Artemieva, 2007]. Let us describe the structure of several database tables for 2-nd level ontology representation. Also, let's demonstrate how this structure is correlated with the terms defined in the 3-d level ontology. Examples are written with means of the applied logic language [Kleshchev, Artemieva, 2005].

"Types of objects" – defines what types of objects ("Subname" field) form the current domain described in the 2-nd level ontology, and how they are represented ("SubsType" field). Value of "Subname" field is the string with the name of an object. "SubType" field can be one of : {}R, {}I, {}N. Representation of information is defined by the 3-d level ontology parameter as the name of the set of terms: sort Types of objects: $\{\}N \setminus \varnothing$.

The view of information of "SubType" field is taken from the description of each element of the set of types as the name of the set : (Type: Types of objects) sort Type: $\{\}(R \cup I \cup N)$.

"Types of objects components" is the table containing the definitions of the types for the objects of the 2-nd level ontology to be created (for each type of object ("Subname" field) it defines what kind of objects will become the components of the current one ("SubsComponent"). This table is linked with the "Object types" table, the values of "Subname" and "Subs Component" fields may be only the types defined in the "SubName" field of "Object types" table. The table may contain several rows with the same "Subname" field value, but the values of corresponding "SubsComponent" field must differ. The representation of information is described with the 3d level ontology parameter as the function, with the input of object type and the output of the set of types:

sort Types of object components: Types of objects $\rightarrow$ {}Types of objects.

"Own properties of objects" is the table containing the names of the sets of own properties of objects for the 2-nd level ontology to be created. This table is linked with the "Object types" table. The value of the "SubName" field may be only the types defined in the "Object types" table. Value of "SubPrivateProp" field is entered by user. In the 3-d level ontology the term "Own properties of objects" is defined as the constructor for the set of functions

[Artemieva, 2007], and the parameter of this constructor is the type of objects: Own properties of objects ≡ (λ(Type: Types of objects) (λ(Area of possible values: {}(Value sets ∪ {}Value corteges)) (j(Type) → Area of possible values)).

“Properties of components” is the table containing the names of the sets of the components of the same type. This table is linked to “Component types” table, the values of “SubName” and “SubsComponent” fields may be only the pairs of values defined in the “SubName” and “SubComponent” fields of the table “Component types”; value of the “SubsComponentsSubsProp” field is the name entered by the user. The term “Properties of components” is also defined in the 3-d level ontology as the constructor for the set of functions [Artemieva, 2007], and the first parameter of this constructor is the object type, second – the set of the components: Properties of components ≡ (λ(Type1: Types of objects) (Type2: Types of objects components(Type1)) (λ(Area of possible values: {}(Value sets ∪ {}Value corteges)) (Object that has type 1 → j(Type1), Objects that has type 2→ Object components(Type1, Type2)(Object that has type1)) → Area of possible values))

“Types of process objects” is the table defining the level of abstraction for the physicochemical process in the domain being defined. In contains the names of objects that may be the participants of the physicochemical process. The information representation is defined by means of the parameter of the 3-d level ontology: sort Types of process objects: {} Types of objects \ ∅.

So the linkage between the database tables corresponds to the relations between the terms of domain metaontology defined by the chemical metaontology. During the creation of the new metaontology of new domain the new database is automatically created and knowledge engineer fills it with the new information. The 1-st level ontology of each domain has the module structure, and each ontology module corresponds to one subdomain. Each module has its own database. 1-st level ontology terms are stored in the table with the following structure: (1) the field for the term of 1-st level ontoogy defining the name of the property (function name); (2) the set this term belongs to (term of the 2-nd level ontology); (3) the arguments of the function; (4) value area of the function.

The value of the third field is defined automatically by defining the name of the set-term of the 2-nd level ontology, because the 3-d level ontology already contains the definitions for each function. The value area of the function may be the set of names, set of integers or real numbers from some interval, the set of structured formulas etc. If the value area is an interval then the table contains the bounds for this interval.

Each set of the graphical objects has its name. Each element of the set corresponds to its editor. This correlation is stored in the special table which contains the names of the graphical editors and the names of subroutine component of knowledge editor for editing this type of objects. Addition of the new editor component is the responsibility of the attendant programmer. Information representation structure in the knowledge base module is defined by means of ontology module (fig. 5).

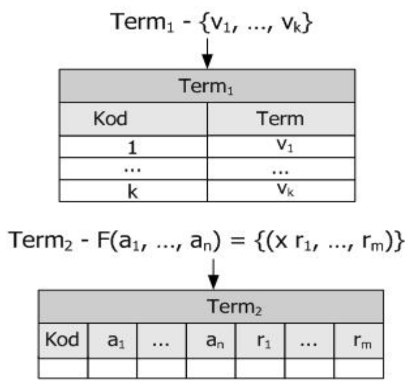


Fig 5. Relation between an ontology term and structure of a database table

Database containing the set of linked tables is automatically created by database control system. The schema of this database is defined in the ontology as the set of terms and their interconnections. If the term is defined in the ontology model as the set, it will be represented in the database as the table containing two fields: unique ID (key field) and the value. If the term is defined as the function, it will become the table where the number of the fields is by one greater (key field) then the sum of arguments number plus the number of elements in the result representation (if the result is not the single value but the Cartesian product then each element of this product corresponds to one table filed). If the result is the predicate then it's regarded as the functions with Boolean result.

The type of each field is defined by means of value restrictions from the ontology module.

Dialogue Scenario of the Editor

Process of creation of the 2-nd level ontology (metaontology of the chemical domain) includes the user-definition of the values of all parameters of the 3-d level ontology. Every constructor of the set defines the scheme of terms belonging to this set. Let's describe the fragment of the dialogue scenario for creating 2-nd level ontology by means of the editor.

1. Define the name of new 2-nd level ontology. In this case the empty database is created, with the same name as the 2-nd level ontology. The scheme of database is based on 3-d level ontology. For example, the name of domain can be "Physical chemistry".

2. Define the names of the sets of objects belonging to this subdomain. The representation of each type of objects must be described as one of available: {}R, {}I, {}N (on other words, there can be three variants of object representations - float or integer numbers, or names). For example, for the "Physical chemistry" domain there can be such types of objects as "chemical substances", "chemical elements", "chemical reactions"

3. Define the structure of objects of each type. On other words, confront to the type of objects the set of other types of objects. For example, the components of the objects of the type "chemical substances" are the objects of the type "chemical elements". The components of the objects of the type "chemical reactions" are the objects of the type "chemical substances" etc. The definition of the components for each type is done by means of choosing the set of available types from the list (as defines at the step 2).

4. Define the terms for labeling the sets of own properties of each object type. In this case editor automatically generates the names for the sets by default; afterwards user can edit these names. For example editor forms the names such as "Own properties of objects that have type <chemical elements>" and "Own properties of objects that have type <chemical substances>". User changes them into the "Own properties of chemical elements", "Own properties of chemical substances". Editor forms all names according to the object type definition of step 2.

5. Define the terms for labeling the sets of properties of object components. In this case editor also automatically generates the default names; and user can edit them. For example editor forms the names such as "Properties of components <chemical elements> for objects that have type <chemical substances>", "Properties of components <chemical substances> for objects that have type <chemical reactions>". User

changes them into the "Properties of chemical elements of substances", "Properties of substances of reactions". Editor forms all names according to the pair definitions <type of object, type of component> of the step 3.

6. The same term can have two or more schemes of definition. This set of such schemes is accepted by the scientist society. During this step user defines for each set or terms its name and the set of possible schemes. This scheme definition step uses the information entered earlier during the editing.

7. Define the terms for labeling the sets of names of relations between the objects of different types, so called the sets of common object properties. In this case editor allows to choose the several object types and to enter the term name. For example user enters term name “temperature-dependent material properties” and defines (by choosing the elements from the list of all object types) that the objects (which are the arguments of this property) belong to two sets “chemical substances” and “tabular values of temperature”.

8. Define the level of abstraction for the physicochemical process. In other words, definition of what object types participate in the chemical processes, are their properties taken into consideration in the chemical processes or not. The user is step by step asked about every object type and its components defined by him/her before. For example, participants of the process in the physical chemistry are chemical materials and reactions between them.

9. Define the terms for common properties of the process and its components. The user has defined the level of abstraction (on step 8), so the editor automatically creates the names for the sets of ontology terms. User also can change these names.

10. The definition of relations between objects “object – its component” leads to the fact that each component can include its own components, and in its turn, subcomponent can also include sub-subcomponents etc. During the investigation of chemical process not only the properties of its direct participant are considered but also the properties of participants’ components are also considered. That’s why the purpose of the next step is to define the depth of such nestling. All relations “object – its component” are already defined, so the editor one by one asks the user, which levels of nestling will be considered in this domain. This step uses the information gathered on step 3 and 8.

11. The 2-nd level ontology is used as the base for creating the 1-st level ontology of subdomain. Creation of 1-st level ontology consists of term definition – representatives for the term sets already defined in the 2-nd level ontology. The meta-term of 2-nd level ontology is used for 1-st level ontology term definition. The 1-st level ontology terms are the names of functions – object properties, their components etc. For each function, the definitional domain is defined by the meta-term, value area is defined by user. User chooses the subset of values from available sets; it can be the set of structured formulas, the set of spectrum etc.

The Fragment of the Second Level Ontology

As an example of editor’s work let’s see the sample 2-nd level ontology for physical chemistry. Let’s start with the definition of values for 3-d level ontology parameters.

1. Types of objects ≡ {Chemical elements, Chemical substances, Chemical reactions}

The ontology defines the objects of the given types. This set is defined on step 2.

2. Types of object components ≡ (λ(Type: {Chemical elements, Chemical substances, Chemical reactions}) (Type = Chemical substances ⇒ {Chemical elements}), (Type = Chemical reactions ⇒ {Chemical substances}), (Type = Chemical elements ⇒ ∅)

Chemical elements haven't components. Components of chemical substances are chemical elements. Components of chemical reactions are chemical substances. This information is defined on step 3.

3. Types of process objects ≡ {Chemical substances, Chemical reactions}

Objects of chemical process are chemical substances and reactions. This set is defined on step 6.

Lets' define the ontological agreements for 2-nd level ontology. They are defined on step 2 when user chooses the way of representation for each object type.

1. Chemical elements ⊂ {}N \ ∅

2. Chemical substances ⊂ {}N \ ∅

3. Chemical reactions ⊂ {}N \ ∅

Ontological agreements of other types can be defined by means of specialized formula editor.

Now lets define the 2-nd ontology terms, and sensible names for constructors.

1. Own properties of chemical elements ≡ Own properties of objects(Chemical elements)

Term "Own properties of chemical elements" means function, which argument is the set of values or the set of corteges of values (m); the result is the function which argument is the chemical element, and the result is the member of set m. This term is defined on step 4.

2. Properties of substances of reactions ≡ Properties of components(Chemical reactions, Chemical substances)

Term "Properties of substances of reactions" means function, which argument is the set of values or the set of corteges of values (m), the result is the set of functions; where the arguments of each function are chemical reaction or its participant (chemical material), and the result of each function is the member of set m. This term is defined on step 5.

3. Properties of elements of substances ≡ Properties of components(Chemical substances, Chemical elements)

Term "Properties of elements of substances" means function, which argument is the set of values or the set of corteges of values (m), the result is the set of functions; where the arguments of each function are chemical material or chemical element, and the result of each function is the member of set m. This term is defined on step 5.

Let's see the example of definition of the 1-st level ontology term, by means of 2-nd level ontology terms.

sort Atomic weight: Own properties of chemical elements (R(0, ∞))

Term "Atomic weight" means function, which argument is the chemical element and result is positive real number.

Knowledge Editor

During knowledge editing user can define only the values allowed by ontology. For example, let's define the term "Current number" as the own property of chemical element. So the definitional domain of this function is the set of chemical elements stored in the table with the same name. Let's define the value area of this function as the integer numbers from 1 to 104. In this case, only the integer number from this range can be assigned as the value of this function for any chemical element.

Another example, let's define the term "Reagents of reaction" as the own property of chemical reaction. So the definitional domain of this function is the set of chemical reactions, and the value area is the set of all subsets of all chemical materials. So for each reaction stored in the table “Chemical reactions” user can choose its reagents from the table “Chemical materials” as he/she defines the values for this function.

If the term is defined as the property of reagent of reaction, then its first argument is the name of reaction, the second one is the name of reagent. In other words, knowledge editor doesn’t allow defining the wrong set of arguments. These restrictions are defined in the metaontology of chemistry.

If the information input about structured formulas is needed, then specialized graphical editor is used (fig. 6). The call of this editor is managed by ontology. Entered by user information about structured formula is automatically transformed into the structured description according to the rules of description in the specialized ontology [Artemieva et al, 2006]. Let’s show the main terms of this ontology.

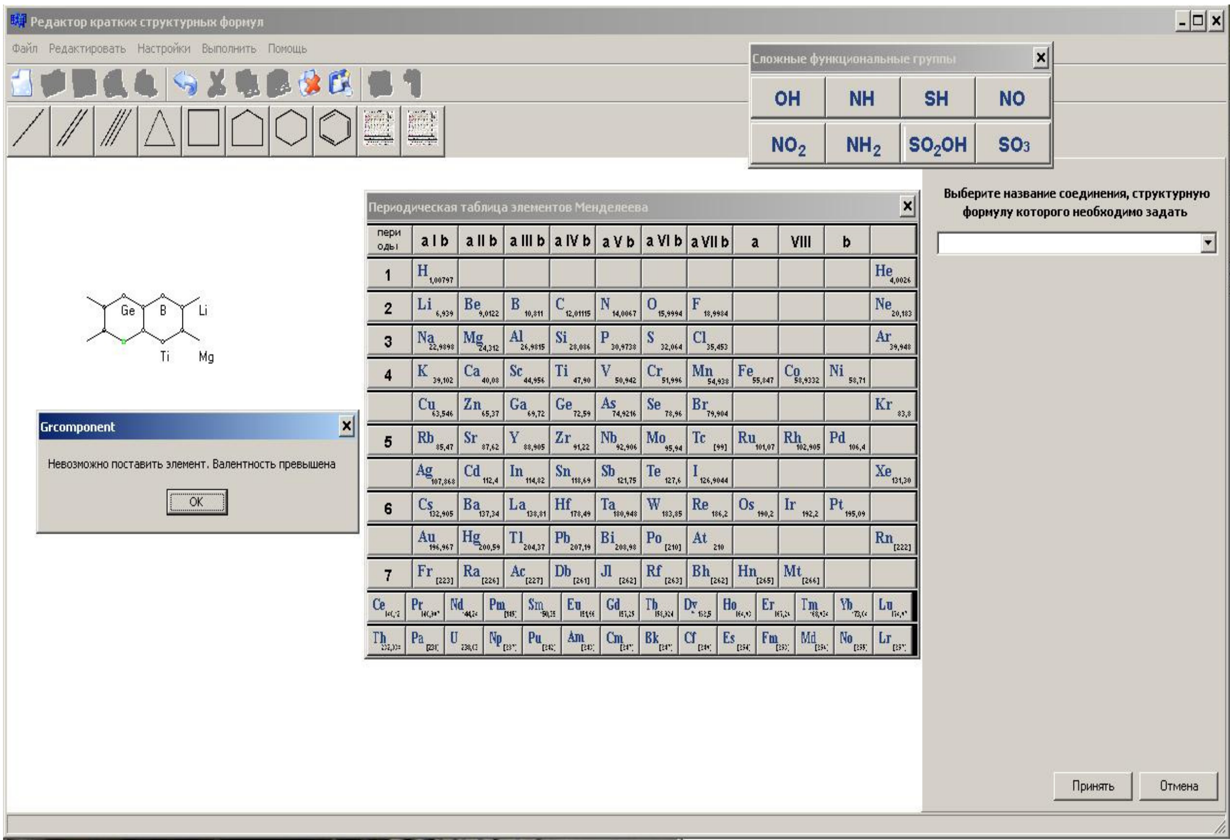


Fig. 6. An interface of the specialized graphical editor for structured formulas

The ontology defines the set of possible types of bond: bond types $\equiv$ {simple, double, triple}.

The structured formula describes the structures bonds of chemical elements with each other, and each element has its own number in the structure. Element numbers $\equiv$ I[1, ∞).

The set of mutual relations between the elements is represented with the term “set of bonds” $\equiv$ ($\cup$ (n:I[1, maximum number of bonds])($\times$ chemical elements, element numbers, bond types)$\Uparrow$ n), which means the set of triple corteges consisting of the chemical element, its number and type of bond. The components of the structured formula are represented as the triple corteges consisting of the chemical element, its number and the set of bonds which this element forms within the structured formula: possible components of the structured formula $\equiv$ ($\times$chemical elements, element numbers, {}set of bonds)). Each structured formula is the sequence of components where the numbers of chemical elements differ for different components: possible structured formulas $\equiv$ {(f: {(n: I[1,∞)) possible components of the structured formula$\Uparrow$ n})(&(i: I[1, length(f)])(&(j1: {(j: I[1, length(f)]) i $\neq$ j}) π(2, π(i, f)) $\neq$ π(2, π(j1, f))))}.

The graphical editor checks all agreements from chemical ontology and knowledge about chemical elements while defining the structured formula [Artemieva et al, 2005], and doesn't allow user defining the values contradicting with the ontology and knowledge. For example, while defining the bond between two chemical elements editor checks, can these two elements with current valencies create this type of bond or cannot.

Conclusion

This paper describes the creation of 3-level ontology and knowledge editor for specialized computer knowledge bank for chemistry. The fragment of chemical meta-ontology is represented. The editor dialogue scenario for creation of chemical domain meta-ontology is shown. The representation structure of ontology and knowledge base by means of database control system is described. The method of adding graphical components to the editor is described. At present time, the prototype of this editor is created; it contains the graphical component for defining the structured formulas of materials.

Bibliography

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

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

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

[Artemieva, 2007] Artemieva I.L., Reshtanenko N.V., Tchvetnikov V.A. Multilevel ontology of chemistry // In the Proceedings of conf. "Knowledge-Ontology-Theory". Novosibirsk, September 14-16, 2007. Vol.1. P. 138-146.

[Artemieva, 2008] Domains with complicated structures and their ontologies // Inf. Theories and Appl. – 2008. - Vol. 15.- № 4. – PP. 330-337. – ISSN 1310-0513.

[Artemieva, Tsvetnikov, 2002] Artemieva I.L., Tsvetnikov V.A. A fragment of physical chemistry ontology and its model // Investigated in Russia.- 2002.- №3.- P. 454-474. (http://zhurnal.ape.relarn.ru/articles/2002/042.pdf)

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

[Denny, 2004] Denny M. Ontology Building: a Survey of Editing Tools // http://www.xml.com/pub/a/2004/07/14/onto.html

[Kleshchev, Orlov, 2006] Kleshchev A.S., Orlov V.A. Computer knowledge banks. The universal way to solve the problem of information editing // Information technologies. 2006. №5. PP. 25-31.

[Kleshchev, Artemjeva, 2005] Kleshchev A.S., Artemjeva I.L. A mathematical apparatus for domain ontology simulation. An extendable language of applied logic // Int. Journal on Inf. Theories and Appl., 2005, vol. 12, № 2. PP. 149-157.

Authors' Information

Irene L. Artemieva *– Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences; 5 Radio Street, Vladivostok, Russia; e-mail:* artemeva@iacp.dvo.ru

EVOLUTION BY CHOICE IN ONTOLOGIES

Stefan Kojnov, Vassil Sgurev

***Abstract**: Blindly copying and mimicking evolutionary processes in animated nature along with well developed mathematical models for them looses its base, the inevitable evolution looses its meaning. Instead the evolution by choice more and more establishes itself and it has much greater perspectives to spread. The experiments of evolution by choice is in an abstract mathematical domain where the models include numerous number of elements, i.e. much too bigger in number than the biological evolutionary models. The paper introduces the idea of evolving-by-choice ontologies, a juxtaposition of evolving ontologies with an evolution by choice algorithm. A principle scheme of evolving-by-choice ontology is introduced and the starting options of the model are presented. In total the fundamental problem for biologically inevitable evolution – prove that the solution space exists – in the case of evolving-by-choice ontologies is replaced by the problem to find the shortest path to the solution space.*

***Keywords**: inevitable evolution, evolution by choice, evolving ontologies, evolving-ontologies state-of-the-art, evolving-by-choice ontologies.*

***ACM Classification Keywords**: C.0 Computer Systems Organization – System architectures, C.4 Performance of Systems – Modeling techniques & Performance attributes, D.2.4 Software/Program Verification - Validation, D.2.8 Metrics – Complexity measures & Performance measures, D.2.9 Management - Life cycle & Productivity, D.2.10 Design – Methodologies, D.4.8 Performance – Modeling and prediction, F.2.2 Nonnumerical Algorithms and Problems – Complexity of proof procedures, G.4 Mathematical Software – Efficiency*

Introduction. Necessity of Evolution by Choice in Ontologies

At present "ontology evolution is barely study" [1]. The undeveloped state of the theory may be treated in two major aspects: (i) shortage and even absence of a general and comprehensive theory due to (ii) blind copying the biological evolution [2] formalized as applying mainly statistical methods from mathematics for phenomena in biology as reproduction, mutation, crossover, etc. [3, 4, 5].

At a closer look the following classification approaches are valid for ontologies: (i) abstract-theoretic fixed at the philosophical concept of ontology with all its typical features and characteristics which include the lowest level of application for a specific domain (the so called formal ontology [6]), (ii) theoretic-applied developing in *principal* implementations in economics (Web services and e-learning [7]); knowledge management and design of information systems [8]; process control systems [9, 10, 11]; (iii) applied for a specific domain: software platforms [12], standards, programming languages and software technologies for e-learning, programming and logical languages, formats and standards for ontologies in management systems.

Dynamics in ontologies is predetermined by the environmental dynamics. It is linked with the temporal properties of the system "object-environment" and it is formalized via temporal dependencies imposing temporal and other limitations. Basic tools to formalize dynamics are the ones of mathematics and of logic, but statistics (statistical analysis systems, statistical packages for social sciences) is excluded.

Recently the research of evolutionary-type changes in ontologies are reduced to establishing inherent concepts and also to their organization in a definite and logically-consistent system. For example some of the basic terms and concepts for the working group in the Karlsruhe University during the last years are: (i) type of evolutionary operations in ontologies [13]; (ii) requirements for ontology evolution [14]; (iii) ontology evolution phases [15]; (iv) types of relations in evolving ontologies [16]; (v) ontology mapping via axioms [17]; (vi) discovery-driven ontology evolution [14]; (vii) evolving ontology evolution [15]; (viii) incremental ontology evolution [16]; (ix) classification of changes in ontologies [15]; (x) revision of ontologies as beliefs' revision [18]; (xi) collaborative engineering of evolving ontologies [19].

Obviously the last more general concepts are directly related to the previous more particular ones. Besides one and the same authors develop and specify both categories of concepts for evolving ontologies. At this stage from the point of view of formalism about evolving ontologies the essence of logical rules (limitations, axioms) undergoes a rapid growth and it removes typical mathematical techniques (inherent to taxonomical and in fact unified approach to processes in a given domain). In this natural way blindly copying and mimicking evolutionary processes in animated nature along with well developed mathematical models for them looses its base, the inevitable evolution looses its meaning [2]. Instead the evolution by choice more and more establishes itself and it has much greater perspectives to spread [2].

Arguments for the latter can be found for example in [15]. The authors introduce the following taxonomy of changes: (i) definition of ontology change (OC); (ii) ontology change subfields (in fact this taxonomy list); (iii) ontology evolution; (iv) ontology versioning; (v) heterogeneity resolution to enhance ontology ensemble operation; (vi) ontology integration; (vii) ontology merging. On the other hand the application of the typical mechanism of evolution for living creatures [20] confronts with invincible contradictions concerning the concepts in this taxonomy except for the concepts of ontology evolution (iii) and its versioning (iv).

In [2] it is postulated that the evolution by choice was an object for experiments in an abstract mathematical domain where the models include numerous number of elements, i.e. much too bigger in number than the biological evolutionary models. The basic advantages of the evolution by choice which are also its identification marks are: (i) the goal formulation does not indicate a respective fitness function; (ii) intermediate solutions and estimates are also an object of the evolution; (iii) processing includes variable reference points rather than probabilistic expertise; (iv) the final result is more reliable compared to the traditional evolutionary approaches. In fact the possible heuristics in the evolution by choice is introduced only by the user; it is replaced by exact knowledge (models or inference schemes); exact knowledge originates during the mathematical proof thus reducing the role of the heuristic information. In contrast to this type of evolution the finalizing procedure of the inevitable evolution is unclear or weak and heuristics is decisive to reduce the solution complexity, etc.

A special matter of interest about the evolving-by-choice ontologies are the concepts for the direction of evolution and for the restrictions of evolution. The directions here may be general and temporal and they may be logically united. The limitations may be static and dynamical. In the case of dynamical limitations the dynamical-restriction weights are much bigger than the weights of static restrictions.

In total the fundamental problem for biologically inevitable evolution – prove that the solution space exists – in the case of evolution by choice is replaced by the problem to find the shortest path to the solution space.

Possible Realization of the Evolution by Choice in Ontologies

The main peculiarity in the case of applying the method of evolution by choice for evolving dynamical ontologies is the following consideration.

The computer component of the human-machine system processes only the syntax of knowledge and the human (the user) is responsible for the semantics and pragmatics of knowledge [2].

The resented below scheme and algorithm of operation in evolving ontologies is based on the presented in [21] (for the ontology scheme) and [22] (for the ontology algorithm) ontology schemes and also to the proposed in [2] algorithm fro evolution by choice.

The Typical Scheme of Evolving Ontologies. Ontology Changes and Ontology Evolution

The basic part in a typical scheme of evolving ontologies is the ontology analyzer [21]. It must fix the change in the object ontology via its comparison with the already accumulated archive about the already fixed ontology changes.

Besides the fixation of the new change and its logging in the library the following new actions are performed: (i) deletion of the already unnecessary class(es); (ii) pooling of some classes, (iii) addition of a new class; (iv) the Universal-Knowledge-Identifier (UKI) Modifier changes the respective UKIs; (v) the ontology version is updated.

Obviously in the most minimal (but *fundamental*) version the ontology evolution is reduced to a fixation and reacting by the human-machine system to the changes in the ontologies and also to the adaptation to them; the *forecast* of the system behavior in the future is *null* and the reconstruction is reduced to reconfiguring the classes *post factum* instead. There is no action concerning *dynamic* parameters (besides the UKIs, classes and the ontology version) and especially the *continuous* dynamics. The changes of UKIs, classes and the ontology version are always triggered by user requests instead of following some law of temporal behavior. The operation in the *temporal* dimension is not possible yet. This means that either the basic evolving-ontology model is *temporally incomplete* or it must be *upgraded towards its temporal dimension*. Therefore the relation between the *system*, the evolving ontology *model* and the application *domain* is vague and also that the environmental *properties* are not reported.

The up-to-date upgrades in the direction of temporal behavior concern just concrete *aspects* of the overall system behavior for the concrete *domain*; the concrete ontologies are also tied to the concrete problem. *This explains the absence of a generalized but unified model of evolving dynamic ontologies*.

In total, the evolution of the *quasi-dynamic quasi-evolving ontology* is *not by choice*. It is even *not inevitable*, it is *unpredictable* instead (except for the user intentions). Such type of evolution resembles to a great extent the evolution of living nature thus obeying the typical for it *uncontrollability* and *unavoidability*.

Evolution-by-Choice. Important Considerations

Following [2] we may split the algorithm of evolution by choice in two big successive parts: pre-evolutionary part and a part most tightly connected to the system evolution.

Pre-evolutionary Part. It corresponds to the formalization of the problem including the goal formulation and the determination of the environmental properties.

These properties are determined via the constraints and/or axioms that are imposed over the system. The environmental properties are available on the basis of the input data. The user inputs a *set of models* with their *priorities*. The input data set is tested for inconsistencies with the *accumulated knowledge*.

The field of the application domain is expanded on the basis of the mathematical induction.

Evolution-by-Choice Part. It adapts the data and/or the knowledge to the available knowledge thus making the model valid also for other possible models.

The mainstay for corrections and changes in the current solutions comprises the process of modeling, fixing and solving contradictions. Therefore the evolutionary process most often changes or it picks up strength along with the system operation.

The most often used techniques include juxtaposition, comparison, grouping and analysis of relations between the data, the knowledge or of their sets.

The *fitness function* is implicit. It is formed and changed.

Initially there is an infinite number of *possible solutions*.

Estimation criteria of the solution are unknown a priori. They are based on the application of “assumption-conclusion” relations of different types which can be within a broad range from completely informal up to Horn rules.

The *best solution* is determined based on the solutions with the highest priorities.

In the algorithm proposed in [2] the *time coordinate* (the temporal dimension) is logically based which in its turn depends on the logical inference. In addition it strongly depends on the *learning* process during the evolution cycle because most of the dynamic changes are unknown a priori.

The power of the algorithm is based on:

- expansion or correction of the accepted *model*,
- expansion or correction of the *space of relations between the input data*,
- reduction of *inconsistencies between the data interrelations and the stored knowledge*,
- ability to ***learn from dynamical constraints***.

The last feature is the cherry on the pie for applying the evolution-by-choice mechanism for dynamical ontologies' evolution. In contrast the learning process in living creatures is inconsistent, chaotic and random while the evolution itself is accidental rather than law-governed.

Comparative Analysis of the Two Introduced Models. Their Unification in Evolution by Choice in Ontologies

In total the basic model from item "The Typical Scheme of Evolving Ontologies" reacts event not right to the evolution stimulating factor itself but rather to the *single changes* in the environment. Consequently such model(s) is (are) contextually a model of a dynamic (or variable) ontology; and formally, from the software point of view – a model of a quasi-evolving quasi-dynamic ontology which responds to the environmental changes.

On the contrary, the model from item "Evolution-by-Choice" is not just an evolutionary model; it is rather a model of a *goal-driven evolution* or of an evolution by choice.

The basis for unifying (and mutual supplement) of those both models may be the presented in [22] scheme of the ontology life cycle. Just like the presented in the "Evolution-by-Choice" item algorithm this scheme may also be divided in two big clusters which are handled one after another: (i) system initialization with data and knowledge which do not depend on the user requirements, and (ii) processes which follow from these requirements.

Entity (i). It comprises the knowledge about the domain, the procedural knowledge, the conceptualization with its explicit specification, the metric characteristics of similarity and the estimates based on them.

Entity (ii). It comprises the factorization of the ontology to shared and private, the contextualization, the ontology development (which includes refinement, generalization, redundancy, similarity), the simplification of the ontology and the creation of a Truth Maintenance System (this block may be considered identical or at least analogous to the mechanism of resolving inconsistencies in the model from item "Evolution-by-Choice").

In addition the state of the ontology can possess several dimensions (temporal, spatial, a degree of similarity and fuzziness as a functional of the affiliation set). It can be determined at least in three possible ways: as a cumulutative network of possible combinations of the formal ontological model and also as a general state – a vector of the partial ontologies. At that each state of every partial ontology is a vector of the components of the formal ontological model.

It is evident that the part closest to satisfying the requirements for an evolution-by-choice algorithm is the one which is responsible for the ontology development with its four basic entries (refinement, generalization, redundancy, similarity). At the same time the formalism describing the ontology state must be supported. The system constraints may be formulated via the tools of logic as axioms and rules of one or several types of logic (descriptive, temporal, etc.).

Principal Scheme of an Evolution-by-Choice Ontology

The scheme of an evolutionary-by-choice ontology is shown in fig. 1.

It can be partitioned briefly in two big parts, before and after the test for changes in the data and in the environment. In its turn the part before the test is subdivided in two smaller parts, before and after the ontology-user-requirements input.

In fact before the test for changes about the ontology there are two *parallel* branches that perform coherently.

System Start-Up Initialization with Data and Knowledge Independent on the User Requirements and Including the Pre-Evolutionary Sequence from Item "Evolution-by-Choice. Important Considerations"

Ontology Life Cycle Model		***Evolution-by-Choice Algorithm***
Domain Knowledge	Operative Knowledge	Goal and Model Input by the User
Conceptualization		Ranking the Solution Candidates
Explicit Conceptualization Specification		Expanding the Interdata Relations (IR)' Space
Metric Characteristics of Similarity		IR Processing
Metric Characteristics Estimates		Search of an Relation between the Goal and the IR

← Ontology User Requirements Input

Choice of Consensus Procedures	Inconsistency Test for IR with the Knowledge
Factorization of Ontology [States] to Shared and Private	Validity Test of IR for Other Models
Contextualization	Applicability of IR for Another Model(s) Test
Refinement, Generalization, Redundancy, Similarity	Test for Other Model(s) with Modified or Partial IR
Ontology Simplification	Test For Other Model(s) Applicability
Creation of Truth Maintenance System	

Test for Change(s) in the Data/Environment? — no → START

Adapting Data and New Knowledge to the Existing Archive:

- ***Ontological Section***: refinement, generalization, redundancy, similarity
- ***Logics' Section***: expanding the set of constraints with a Simultaneous Growth of the Number of Logical Models (see item "Evolution-by-Choice. Important Considerations")
- ***Programming Section***: Deletion of Unnecessary Class(es), Juxtaposition of Class(es), Addition of New Class(es), Validation of UKI Modifier(s), Editing Ontology Version (see item "The Typical Scheme of Evolving Ontologies. Ontology Changes and Ontology Evolution")

START

Fig. 1. Principal Scheme of an Evolution-by-Choice Ontology

Before the ontology user requirements input the first branch (it is denoted in the left half of the scheme) is responsible for the basic types of initialization of the ontology at first or for the iteration (domain and operative knowledge, conceptualization, explicit conceptualization specification, metric characteristics of similarity and metric characteristics estimates). Simultaneously with it the second branch (in the right half of the scheme) is responsible for the startup initialization of the logic tools (input of the goal and the model by the user, ranking the solution candidates, expanding the interdata relations (IRs) space, IR processing, search of a relation between the goal and the IR).

After the input of the user requirements for the ontology on the left side the processing continues with the choice of a consensus procedure, followed by the factorization of ontology [states] to shared and private ones, contextualization, refinement, generalization, redundancy and similarity, ontology simplification, and the creation of a truth maintenance system. The logical tools proceed with an inconsistency test for the interdata relations (IRs) with the accumulated knowledge, followed by a validity test of IRs with other models, with an applicability check of IRs for another model(s), with a test for other model(s) with modified or partial IRs, and with a test for applicability of other model(s).

The adaptation process of the ontology to the changes in the environment and/or in the model are mainly grouped to three main phases: (i) for the left half of the scheme, a development of the ontology (refinement, generalization, redundancy and similarity), (ii) for the right half of the scheme, expanding the set of constraints with a simultaneous growth of the number of logical models (see item 2.2) and (iii) deletion, juxtaposition and addition of new class(es), validation of the UKI Modifier(s), and editing the ontology version (see item 2.1). It is evident that (i), the first phase, concerns the evolution of the ontology itself or the *ontological evolution*, that (ii), the second phase, is directly connected with the *logical evolution* of the system and that the third phase, (iii), is connected with the *algorithmic-programming evolution* of the system. So we may conclude that the system evolution is deployed to three successive [branches of] evolutions (ontological, logical and algorithmic-programming).

In fact this partitioning of the overall algorithm to three successive evolutions is rather conceptual than crisp, it marks three substantially different kinds of processing from which the first two concern the internal system mechanism (or modes of operation) on a *semantic* level and the third one – on a *formal* level (the ontological and logical evolutions are formalized via programming constructions).

The mutual coordination between the two semantic aspects of the system evolution by choice is substantial for the good performance of the system as a whole. Kinds of preferable types of logic are the descriptive and temporal logic, the paraconsistent logic and so on.

Due to the extreme novelty of the proposed method the system *starting version* is the minimal possible on. It shall be upgraded in the course of time developing towards its gradual sophistication.

The following options have been accepted for the starting version of the system:

- implementation only of a subset of temporal logics including the relations ‘preceding’/’before’ and ‘succeeding’/’after’;
- system of priorities for achieving the goal ranging from ‘low’ through ‘middle’ up to ‘high’. This system of priorities resembles to a great extent the apparatus of fuzzy logic (though it is *not* a true fuzzy logic at the beginning; subsequently the authors have the intention to apply a simple type of fuzzy logic);
- Prolog-like syntax for inference of new constraints corresponding to the new changes for system operation.

Conclusion

Blindly copying and mimicking evolutionary processes in animated nature along with well developed mathematical models for them looses its base, the inevitable evolution looses its meaning. Instead the evolution by choice more and more establishes itself and it has much greater perspectives to spread. The experiments for evolution by choice is in an abstract mathematical domain where the models include numerous number of elements, i.e. much too bigger in number than the biological evolutionary models. The basic advantages of the evolution by choice which are also its identification marks are: (i) the goal formulation does not indicate a respective fitness function; (ii) intermediate solutions and estimates are also an object of the evolution; (iii) processing includes variable reference points rather than probabilistic expertise; (iv) the final result is more reliable compared to the traditional evolutionary approaches. The paper introduces the idea of evolving-by-choice ontologies, a juxtaposition of evolving ontologies with an evolution by choice algorithm. A principle scheme of evolving-by-choice ontology is introduced and the starting options of the model are presented. In total the fundamental problem for biologically inevitable evolution – prove that the solution space exists – in the case of evolving-by-choice ontologies is replaced by the problem to find the shortest path to the solution space.

Bibliography

[1] N.Guelfi, C.Pruski, Ch.Reynaud, Understanding and Supporting Ontology Evolution by Observing the WWW Conference. International Workshop on Emergent Semantics and Ontology Evolution ESOE, Busan - Korea, November 2007, pp. 19-32.

[2] V.Jotsov, Evolutionary parallels. Proceedings of the First International IEEE Symposium "Intelligent Systems", Varna, Bulgaria, September 10-12, 2002, pp. 194-201.

[3] J.Koza, Genetic Programming. In: On the Programming of Computers by Means of Natural Selection, The MIT Press, Cambridge, MA, 1992.

[4] W.Kruiskamp and D.Leennaerts, DARWIN: CMOS Opamp Synthesis by Means of a Genetic Algorithm. Proceedings of the 32nd Design Automation Conference ACM, New York, 1995, pp. 433-438.

[5] P.N.Surry and N.Radcleff, Formal Algorithms + Formal Representation = Search Strategies. In: Parallel Problem Solving from Nature IV, LNCS, Springer-Verlag, Berlin etc., 1966, pp. 336-375.

[6] V.Petrov, Process Ontology as an Expression of the Idea of Dynamism in Philosophy. Proceedings of SOLON 2007. Sofia, 2008.

[7] Moodle – A Free, Open Source Course Management System for Online Learning. Available at: http://moodle.org.

[8] K.Kurbel and I.A.Loutchko, Framework for Multi-Agent Electronic Marketplaces: Analysis and Classification of Existing Systems. Proc. of the Internat. NAISO Congress on Information Science Innovations (ISI'2001), Dubai, U.A.E., March 17-21, 2001, http://www.wiwi.euv-frankfurt-o.de/wiwww/de/research/project/fp_softcomp_veroeff_pdf/Anlage 203.pdf.

[9] L.Carrilo, J.Marzo, P.Vila and C.Mantilla, MAantS-Hoc: A Multi-agent Ant-based System for Routing in Mobile Ad Hoc Networks, 2005.

[10] M.Dorigo, G.Caro, Ant Algorithms for Discrete Optimization, 2004.

[11] M.Hadjiski, V.Boishina, Dynamic Ontology-based Approach for HVAC Control via Ant Colony optimization, DECOM'2007, Izmir, Turkey, 2007.

[12] Ontologies: Methods, Tools, Applications, IIT-BAS Ontology Seminar, Sofia, Bulgaria, 2006-2007.

[13] G.Flouris, D.Plexousakis, Handling Ontology Change: Survey and Proposal for a Future Research Direction, TR-362, FORTH-ICS, September 2005.

[14] S.Castano, A.Ferrara, G.N.Hess, Discovery-Driven Ontology Evolution, 2006. Available at: http://www.boemie.org/files/Paper_SWAP2006_UNIMI.pdf.

[15] G.Flouris, D.Plexousakis, G.Antoniou, Evolving Ontology Evolution, 2006. Available at: http://www.ics.forth.gr/isl/publications/paperlink/fgeo_SOFSEM06_IT.pdf.

[16] P.Haase, and Y.Sure, D3.1.3 Incremental Ontology Evolution. In: EU-IST Integrated Project (IP) IST-2003-506826 SEKT, Institute AIFB, University of Karlsruhe, 2007, Available at: http://www.aifb.uni-karlsruhe.de/WBS/pha/publications/sekt_d312.pdf.

[17] M.Ehrig, and Y.Sure, Ontology Mapping by Axioms (OMA). In: Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2005, pp. 560-569.

[18] M.Mazzieri, and A.F.Dragoni, Ontology Revision as Non-Prioritized Belief Revision, International Workshop on Emergent Semantics and Ontology Evolution, Ontology Revision as Non-Prioritized Belief Revision, International Workshop on Emergent Semantics and Ontology Evolution (ESOE-07), Busan, South Corea, 12/11/2007. pp. 58-69.

[19] G.A.Vouros, K.Kotis, Ch.Chalkiopoulos, N.Lelli, The HCOME-3O Framework for Supporting the Collaborative Engineering of Evolving Ontologies, ESOE 2007 International Workshop on Emergent Semantics and Ontology Evolution, ISWC 2007, November 12th, Busan, Korea. Also in on-line Proceedings of CEUR-WS.org/Vol. 292.

[20] V.V.Emel'anov, V.M.Kouraychik, V.V.Kouraychik, Theory and Practice of Evolutionary Modelling, Moscow, Fizmatlit, 2003 (in Russian).

[21] D.Rogozan, G.Paquette, Semantic Annotation in e-Learning Systems based on Evolving Ontologies, SW-EL'05: Applications of Semantic Web Technologies for E-Learning, AIED'05 workshop, Amsterdam, 2005, pp. 61-64.

[22] Dynamic Ontologies in Multiagent Information and Control Systems, Report for the Contract between UCTM and NSF in the Ministry of Education ВУ-ТН 204/2006, No. 821 for UCTM, First Year, 2007, Sofia, Bulgaria (in Bulgarian).

Authors' Information

Stefan Kojnov *– Researcher, Institute of Information Technologies, BAS, Acad. G.Bontchev St., bl.29A, P.O. Box: 161, Sofia-1113, Bulgaria; e-mail: slk@iinf.bas.bg*

Vassil Sgurev *– Academician; Institute of Information Technologies, BAS, Acad. G.Bontchev St., bl.2, P.O.Box: 161, Sofia-1113, Bulgaria; e-mail: sgurev@bas.bg*

EXPERIENCE OF DEVELOPMENT AND APPLICATION OF EXPERT SYSTEMS FOR ECONOMIC MACROPARAMETERS FORECASTING AND OPERATION

Alexey Voloshyn, Victoria Satyr

Annotation: *In this article there are considered problems of forecasting economical macroparameters, and in the first place, index of inflation. Concept of development of synthetical forecasting methods which use directly specified expert information as well as calculation result on the basis of objective economical and mathematical models for forecasting separate "slowly changeable parameters" are offered. This article discusses problems of macroparameters operation on the basis of analysis of received prognostic magnitude.*

Key words: *decision tree method, index of inflation, expert information, operation.*

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

Introduction

In articles [Voloshyn, 2003, 2005, 2006, 2007, 2008] presented on "Knowledge-Dialog-Solution" (KDS) conferences there is developed concept of "subjective multideterminism" [Voloshyn 2006, 2007] for qualitative forecasting of economic macroparameters and in the first place of inflation index. This concept is based on the fact that effect is determined by the variety of interdependent reasons (objective and subjective). Reasons themselves (factors, parameters) and intensity of their interaction are determined subjectively (by expert judgment) and are presented by "illegible" decision tree (by its tops and arcs appropriately), "effect" (result of forecasting) is resented by leaves of the tree. There are developed instruments that allow to execute tree derivation (to insert and to eliminate tops, specify rating of arcs), to find more probable development of scenario of predictable process, to define magnitude of predicting parameter "illegibly". The construction of an applied system of support of decision-making is reduced to highlighting by experts of problems and sub problems (tops of the tree) and links between them (arcs of the tree). Experts determine the weights (probabilities) of transitions between tops. It is acceptable to get unclear expert estimations achieved by the Boolean variables described by the values of function of belonging (by vectors of the real numbers from 0 to 1). Every expert sets three estimations – optimistic, realistic and pessimistic, the scaling of which is effected taking into account the psychological type of the expert. The type is determined on the basis of psychological tests built into the system. Such psychological tests help to determine the coefficients of «veracity», «independence», «caution», etc.

The tree is built on the basis of collective estimations of experts using the method of pair comparisons. The construction of a resulting tree requires the use of the algebraic methods of treatment of expert information, and the Hemming metrics and measure of lacks of coincidence of grades of objects is used as distance between ranges. A resulting tree is determined as the Kemeny-Snella median or as a compromise [Voloshyn, 2005]. In the case of setting priorities in an unclear form the elements of matrix are set through the functions of belonging.

The algorithms of successive analysis of variants [Voloshyn, Panchenko, 2002] allowing to process trees with hundred tops are offered for determining optimum ways within the tree.

At KDS conferences there were voiced and published in [Voloshyn, 2007] predictable magnitude of inflation index I Ukraine diverging from actual for 3-5%.

Particularly in June of 2007 at KDS-07 inflation index on 01.01.2008 was set calculated in the first quarter of 2007: “National Bank of Ukraine predicts inflation in Ukraine in 2007 at the rate of 7%, government – 8%. Our prediction is 17.3%. We just need to wait for the beginning of KDS-08 conference and to compare our predictions”. As it’s known actual rate Of inflation was 16.6%. If to take into account that inflation index was considered by us as “unclear” parameter and magnitude 17.3% - “the most probable” (maximizing membership function) received result is worth attention. And with previous “coincidences” mentioned above to the authors’ opinion it is worth scrupulous attention.

At present (April, 2008) inflation in Ukraine is topic number one. For the first quarter of 2008 inflation in Ukraine has become the highest amongst all Former Soviet Union countries (CIS), specified in budget magnitude of 9.6% are already exceeded (9,7%). Cabinet of Ministers of Ukraine has worsened forecast of inflation growth according to the results of 2008 from 9, 6% to 15,3% which is mentioned in the Cabinet of Ministers of Ukraine Resolution No. 581 dated June 27, 2008 “Adoption of Amendment to the Cabinet of Ministers of Ukraine Resolution No. 976 dated July 27, 2007”, posted on the site of the government as UNIAN broadcasts. The government also changed forecast of price index for industrial production manufacturers according t the result of that year from 19, 2% to 24, 6%. As it was notified in accordance with data of State Statistics Committee inflation in Ukraine from May to April was 1, 3% by May, 2007 – 31, 1%. In April, 2008 by April of the previous year it was 30, 2%, in January-April till analogous period of 2007- 24, 4%. From the beginning of current year till April growth of consumer prices was 13, 1%.Inflation in April contrary to that in March, 2008 was 3,1%. Growth of price index for manufacturers in Ukraine in January-May, 2008 contrary to January-May, 2007 was 31, 7%. According to data of State Statistics Committee growth of manufacturers’ prices in May contrary to April, 2008 was 3, 7% and to December, 2007 – 24,2%. Cabinet of Ministers of Ukraine improved forecast of nominal GDP according to the results of 2008 from 921,2 to 956,8 billion hryvnas without change of its growth at the rate of 6,8%.

Operation of predictable parameters

While controlling socio-economic processes especially in the period of transition it is necessary to foresee inflation rate alongside with other factors. Inflation substantially and sometimes vitally influences on flow of real income of population, formation of investment climate, pace of industry and agriculture development, execution of budget’s income, exchange rates and so on.

Forecasting models we have review earlier [Voloshyn, 20

03, 2005, 2006, 2007] can be referred to the class of “Positive” that answer the question “What will be?”. Though experts evaluating degree of parameters’ interaction to some extent take into consideration its change at forecasting interval, these models can be considered ”time-independent” (or stationary). “Normative” models answer the question “How shall it be?” and must recommend amending “stationary” scenario. In our models it is realized by the following reformations in “decision-making tree” - by determination of “narrow places” (arcs and tops that down to the limit influence on the result) with the purpose of recommendation of “correction” of causes degree interaction (in particular inserting and eliminating tops) for receiving desirable magnitudes of forecasting parameter (for example, value pf inflation index mustn’t exceed 10%”). In addition, suggested models and program instruments of their realization accordingly differ from suggested in the previous realizations at the following point. For “objectification” [Voloshyn, 2006, 2007] of decision-making process in predictable model some ratings of arcs are determined on the basis of relative economical and mathematical models (see further). From formally mathematical point of view “correction” of decision-making tree is analysis of its sensitivity [Voloshyn, 2006] but if correction of arcs’ rating can be called traditional analysis insertion-elimination of tops needs development of ad hoc algorithms.

Forecasting and operation of inflation in “bottlenecks”

Hundreds of monographs and thousands of articles are devoted to the research of inflation. We will shortly describe main factors effecting inflation that were used by us in a varying degree for forecasting of inflation rate [Barro, 1998], [Greene, 2000], [Baumol, 2001], [Poroshenko, 2008].

First, worldwide tendency of the last years is impetuous increase in price for supplies and energy resources. Index of world prices for supplies that is calculated by investment bank Goldman Sachs in 2006 grew for 26%, and in 2007 for 41% in the connection with what bank’s experts introduced the term of “agflation” (sudden increase in price for agricultural production).

Second. Increase in price for fuel is also long-term tendency. On 16th of April, 2008 oil price broke another record achieve USD 115 for barrel.

Third. Also more serious problem in Ukraine is impetuous expansion of foreign liabilities of bank. Only for 2007 amount of drawn by banks foreign credits grew almost twice – from USD 14 to USD 31 billion. As these means are used for consumer lending such debt is becoming powerful factor of inflation.

Forth. Inflation wouldn’t be so noticeably sped-up if economy of Ukraine responded key criteria of market economy. In particular there are not used possibilities for progressive expansion of labour productivity.

In the purpose of “objectiveness” in modified variant of expert system for forecasting parameters a priori change of which isn’t spasmodic (for example, in demography, see figure 1) there are used regression models.

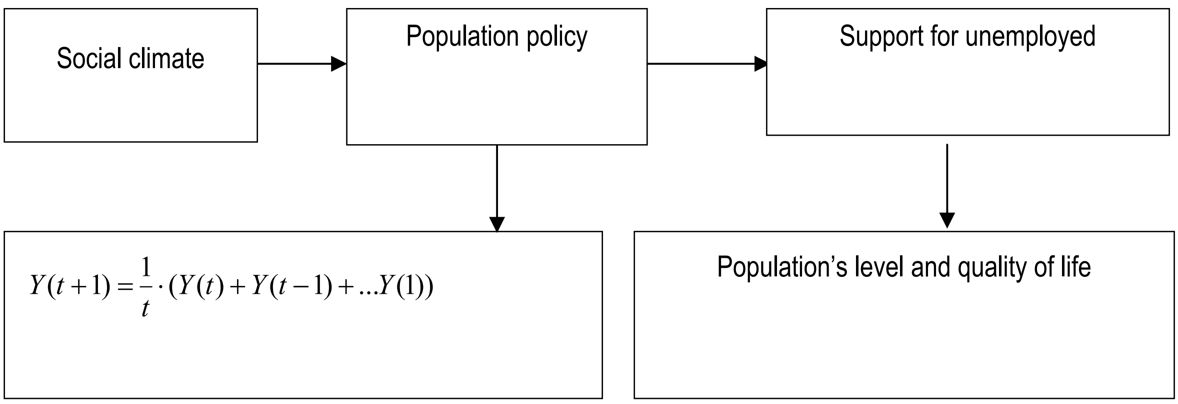


Figure 1.

Preliminary tests of models demonstrated that forecasting for trend equations doesn’t present any correct predictions because dynamics of inflation rate changes is difficult. That is why single-factor regression models were chosen. Multiple factors should be taken into consideration during forecasting including government, bank and financial structures policy, real changes in economy, world markets situation, position of developed countries governments concerning Ukraine sand many other factors. Regression equation looks like Y=a+bx, where Y is composite Consumer Price Index in percentage by previous December (predictable rate); a- fragmenton equal to initial value of dependent variable; b – inclination that shows how much inflation rate is changing with the change of factor x per unit.

The following regression models are used. The simplest model based on plain averaging is

$$Y(t+1)=(1/(t))*[Y(t)+Y(t-1)+...+Y(1)],$$

and unlike the simplest “naïve” model which is conformed with “tomorrow it will be as it’s today” principle this model is based on “tomorrow it will be as it is today for the latest time”. Such model is for sure more resistant to fluctuations because mavericks about the mean are smoothed. However, this method is as simple ideologically as “naïve” models and it has almost the same defects. In the above presented formula it is supposed that row is averaged during quite long period of time. However, as a rule values of time series from the recent times characterize forecast than “older” values of this row. Then moving average $Y(t+1)=(1/(T+1))*[Y(t)+Y(t-1)+...+Y(t-T)]$ can be used for forecasting. Its meaning is that model sees only recent past (per T time counts deep) and being based only on these data it makes prediction. For forecasting exponential mean-value method was also used. Formula describing this model looks like $Y(t+1)=a*Y(t)+(1-a)*\hat{}Y(t)$, where $Y(t+1)$ is a forecast for the following period of time; $Y(t)$ – operational significance at the period of time; $\hat{}Y(t)$ – previous forecast at the moment t; a – consonant of evening-out $(0<=a<=1)$.

Inflation influences not only on debt relationship of such subjects like households, bank, entrepreneurial and financial sectors but also on debt relationship between state and households and entrepreneurial sector (national domestic debt) and intergovernmental debt relationship. Consequently, the next connection we are going to consider will be inflation and national debt. Reasons of limited capacities of national debt loss of value in the modern conditions are:

- Increase of state short term debts portion;
- Increase of sensitivity rate (elasticity coefficient) of reaction of interest rate value on inflation rate;
- Provision with long-term and intermediate credits not in corpore at once but by tranches that allows considering inflation processes influence on loan amount.

There were considered influence of inflation on long-term loans. Let’s discuss influence of inflation on contracts that determine salary level. Labor market supply and demand have common determinant (according to the neoclassical theory) which is actual salary level. Agreeing on its amount, employers and employees react in a definite way on conditions that have been formed at the labor market. If, for example, gross domestic product rate (GDP) and employment are high salary tends to increase. And vice versa – if GDP and employment decreases salary is increasing slowly. Phillips’ wage curve represents connection between salary inflation (loss of value) and GDP decelerating and can be expressed by formula : $q=\lambda(Y-Y^*)$, where λ is sensitive coefficient (elasticity) of salary response on changes of GDP amount, Y – GDP real volume, Y^* - potential.

Taking into account inflationary expectations equation of Phillips’ curve changes into $q = Pe + \lambda(Y-Y^*)$, where Pe - expected inflation rate. There is presented fragment of mathematical model built in decision tree on the figure 2.

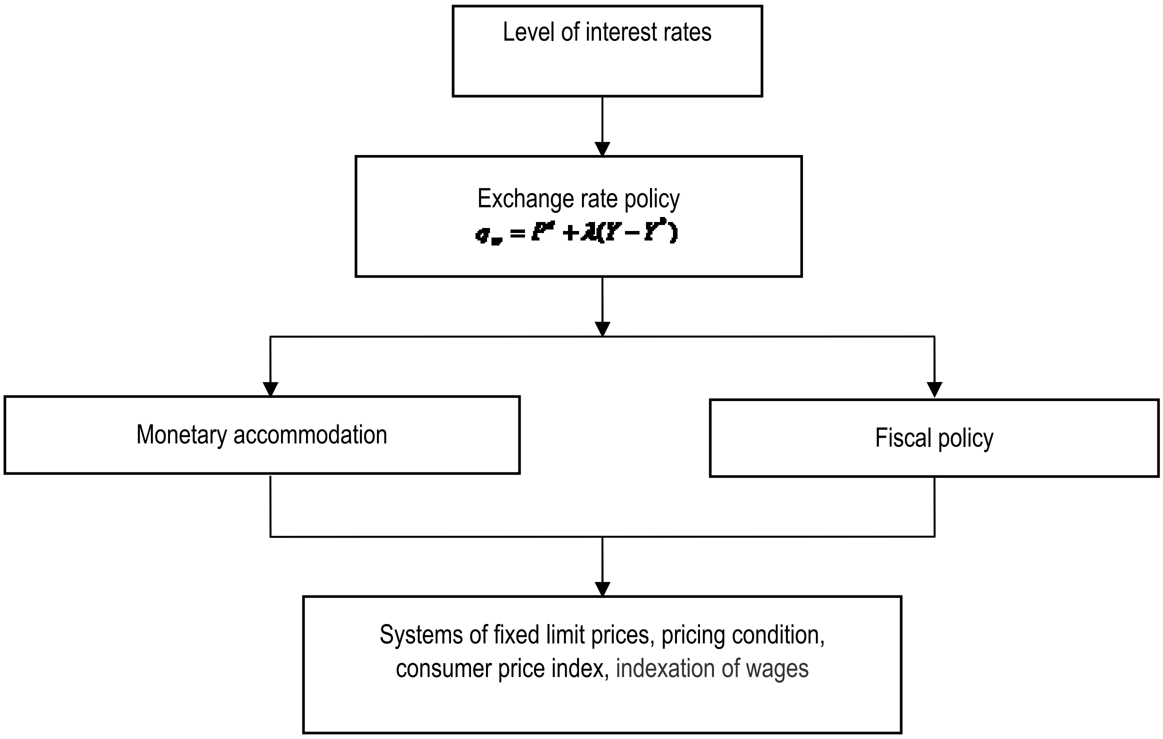


Figure 2

The last equation means that at any prescribed GDP level salary increases faster than anticipated inflation rate. It is supposed that nominal wage increases at 1% faster for every additional percent of anticipated inflation.

Said about demands answer to the question “Why do enterprises agree for such salary acceleration?” Such behavior of entrepreneurs is explained by the fact that increase of nominal wage will not be vulnerable for them if prices for their production are growing at the same rate. In such case both entrepreneurs and employees are in such conditions in which they would be if there is no inflation.

Consequently, wage-push expects anticipated inflation. But as we already know inflation can be unexpected. In such case employees realize wage-push and inflation of their salaries only after some time and will demand relevant compensation. In such situation compensation becomes recovery of losses only in case of unexpected inflation that wasn’t taken into consideration in previous labor contract. Collective arrangements entered by trade unions that have requirement of wage indexing with adjustment for inflation also bring particular confusion on the matter if this compensation is for previous inflation or the expected one.

This condition is very important for the further forecasting of inflation rates. If labor rates for the next year expresses inflation rate of the previous year and prices are based on the actual salary so today inflation will express yesterday’s one and inflation rates will change slowly.

- If for determining labor rates only anticipated inflation is taken into consideration the policy radical change is possible that changes expectation concerning fast change of inflation rate. Besides mentioned aspect it is important to remember another aspect of this problem. Salary can be indexed for 100% or only partially for example for 50-60%.

- There are two ways of salary partial indexation:
- Upper bound delimiting of prices increase exceeding of which provides compensation. For example, if inflation rate exceeds 5% so from 5,1% income begins to be indexed;
- Upper bound delimiting of compensation. This way limits values that are used for compensation of prices increase by beforehand fixed percentage. Making labor contracts for 2-5 years can't provide with complete certainty change of inflation rate especially in shaky economy. That is why for adjustment for inflation there are used two the most widespread ways:
- indexation of wages in accordance with Consumer Price Index and periodic (quarterly or once a half of year) salary revision regarding its increase according to the prices increase for this period;
- prediction of periodic salary increase announced beforehand considering increase of expected prices rates.

If inflation could be predicted with complete certainty both methods would have similar consequences. But as expectations are often mistaken it is considered that indexation based on actual rates of inflation guarantees stability of actual salary more reliably then when pay-outs are decided beforehand.

Inflation influences not only on share of debtors and creditors, employees and entrepreneurs but also it effects state interests. Yes, in conditions of inflation there occurs voluntary prolongation of payments for state budget by taxpayers that allows taxpayer to settle accounts with state by undervalued money. This phenomenon of inflationary taxing in scientific literature was called "Oliver-Tanzy effect".

This effect shortly can be explained as following: any inflation decreases tax weight. It develops with more strength with growth of inflation rate and prolongation of tax payment time.

At the same time it must be noticed that inflation weakening tax weight predetermined by explained above reasons creates another, opposite tendency.

Forecasting of inflation can be done on the basis of index of consumer price change with consideration of pent-up demand. Pent-up demand is equal to population compulsory savings. They are calculated by progressive total for range of years taking into consideration of their year-on-year increase. At the same time index of required savings is equal to good circulation and services index. Year-on-year increase and good circulation, and services value ratio present hidden inflation or population cash income index and good circulation and services index ratio.

On the macrolevel inflation rate can be determined on the basis of Fisher's equation.

$$MV = PQ,$$

where M is amount of currency in circulation; V - velocity of money; P - price; Q - good sales and services value.

As inflation is caused by many factors, its level can be forecasted by development of multifactor models. As factors there may be: change in the exchange rate, growth of money supply, change of refinancing rate and so on. Popular way of forecasting inflation is also calculation of its level on the basis of GDP deflator. Thus inflation shall be explained as money's loss of value. One of the most obvious factors of inflation presence or absence, its depth is price index. As indirect indicator of inflation level are used statistic data of trade stock and amount of population money on deposit ratio (reduction of stock and increase of deposits give evidence about increase of inflationary tension rate). Forecasting of inflation is made on the basis of index of consumer prices change with taking into consideration of pent-up demand. On the macrolevel inflation rate can be determined on the basis of Fisher's equation.

Conclusions

In the development of qualitative forecasting concept on the basis of unclear decision tree it is offered to use "built-up" economical and mathematical models of forecasting separate parameters in the first place determining tree's "bottlenecks". On the basis of the previous analysis of received prognostic values of inflation index in Ukraine for 2008 there are suggested the following (in first priority decrease order) controlling influences with the purpose of correction of adverse forecast of inflation index values (our previous forecast is rated at 30%): salary indexation of budget, graduated income tax levy, backing of agricultural sector, state control of monopolies ("natural" and "artificial"), introduction of minimum hourly pay, reduction of interest for bank loans, refusal from pegging of national currency to USD.

Bibliography

[Voloshin, 2003] Voloshin O.F., Panchenko M.V. The System of Quality Prediction on the Basis of a Fuzzy Data and Psychografphy of the Experts. International Journal "Information Theories & Applications", 2003, Vol.10, №3. -P.261-265.

[Voloshin, 2005] Voloshin O.F. Problems of Decision Making in Socio-Economic Systems// «KDS-2005» conference works, Vol. 1, Sofia, 2005. - p.205-212.

[Voloshin, 2006] Voloshin O.F. Decision Support Systems as Personal Smart Tools of Decision-Making Person // «KDS-2006» conference works, Sofia, 2006. - p.149-153.

[Voloshyn, 2007] Voloshyn A, Satyr V. Problems of Forecasting of Economic Macroparameters // «KDS-2007» conference works, Sofia, 2007, vol. 1.-p.264-269.

[Voloshyn, 2007] Voloshyn A, Satyr V. The inflation Index Prognosis Based on the Method of Decision-Making "Tree"// "Informations technologies & application", 2007, vol.14, N1.-P.63-67.

[Voloshyn, 2008] Voloshyn A, Satyr V.. Experience of Development and Application of Expert Systems for Economic Macroparameters Forecasting and Operation // «KDS-2008» conference works, Sofia, 2008, Vol. 2.-C.1-6.

[Barro, 1998] Barro R. Macroeconomics. – London: The MIT Press Cambridge, 1998. -867 p.

[Greene, 2000] Greene W. Econometric Analysis. – New York: Prentice Hall, 2000.- 1004 p.

[Baumol, 2001] Baumol. What Alfred Marshal Didn't Know: Contribution of XX Century to Economics. – Economic Issues. – № 2. – 2001. – p. 73-107.

[Poroshenko, 2008] Poroshenko P. Antiinflationary Policy: Shock or Therapy? // «Zerkalo Nedeli», 2008, №15.

Authors' Information

***Oleksiy Fedorovych Voloshyn**, Kyiv Taras Shevchenko National University, Faculty of Cybernetics, Professor. Kyiv, Ukraine. E-mail: ovoloshin@unicyb.kiev.ua*

***Victoria Valeriyivna Satyr**, Kyiv Taras Shevchenko National University, Faculty of Cybernetics, postgraduate student. Kyiv, Ukraine. E-mail: Brili@Ukr.net*

MULTIAGENT SYSTEM FOR SMART DOCUMENT ANALYSIS

Vyachslav Lanin, Elena Mozzherina

***Abstract**: In the article you can find an intermediate outcomes for the complex approach to implementation of development electronic document management subsystem in the CASE-system METAS. The system meant for creating distributed information systems that allow dynamic settings for changeable environment and user requirements. The operational efficiency of electronic documents processing is suggested to increase considerable because of automatic smart document analysis. Agent- and ontology-based approaches are suggested as a solution. Ontologies are used to present the semantic and the structure of the document explicitly. Agents are used for simplification of analysis, made it scalable and upgradable. The results of smart analysis and processing of the documents taken from the heterogeneous sources can be used not only for automatic document classification and clusterization in information system in user friendly form, not only for reducing the labor intensity for subject field analysis and design of information system, but also for intellectualization the processes of reports creating based on information from the database of the system.*

***Keywords**: ontology, agent, multiagent system, smart search, document analysis, adaptive information systems, CASE-technology.*

***ACM Classification Keywords**: D.2 Software Engineering: D.2.2 Design Tools and Techniques – Computer-aided software engineering (CASE); H.2 Database Management: H.2.3 Languages – Report writers; H.3.3 Information Search and Retrieval – Query formulation.*

Introduction

Nowadays a large number of CASE-systems, which automate the most labor-consuming stages, which connected with the business flows and interface creation, in Information System (IS) development, exist. The stage of the subject area analysis, which is not usually automated by the CASE-systems, becomes the longest and the most labor-consuming. Thus, one of the most promising trends in the CASE-system development is the automation of this process. This task becomes especially vital in the CASE-systems, which are oriented for the creation of IS with the dynamic adaptation during their usage, and where the stage of the subject area analysis lasts for all the time when the system is used. Usually during the usage of such systems the task of development is entrusted (even partly) to the users, which are experts in the subject area but not in the programming, so components for the automation analysis become the most important. In other words, if we pose the problem of the dynamic setup of IS in changeable environment, then the basis of creating the tools for its dynamic adaptation become the tools for restructuring the data in the Data Base (DB). And these tools allow making changes in the data model based on the results of the subject area analysis, on the normative reference and administrative documents, which regulate the activity in this area. Hence follows the need for support the stage of analysis in dynamically adapted systems, which is one of the most complex and labor-consuming stages in the IS development. As information source for the analysis the documents of different types can be served, because the activity of any business-system is build on the normative documents. The support of the business with the help of the IS tools requires the reflection in the data model the system of the standards, fixed in the normative-reference and administrative documents, in the form of limitations (attributes, subject area objects' properties and relations between them) and operations, assigned on the data [1].

As a result of the analysis the system of the interconnected document will be constructed:

- the documents, that belong to the determinate directions of the business activity (to the specific concepts, to the objects of the subject area);
- the documents, that reflect the relations between these concepts (one or several documents can be connected with each concept and the connections between documents reflect the relations between the concepts);
- the documents, that contain the normative information, which also can be extracted with the help of the documents content analysis.

On the basis of the constructed system of the interconnected documents it is possible to partially automate the process of the analysis of changes in the subject area and introduction of changes in the subject area model (i.e. to implement the support of the development process and adaptation of IS). Thus the document management system becomes not only the «wrapper» above the IS and its DB, that allow to view the data processing results, that are stored in DB, in a user-friendly form, but also becomes the basis of the IS development tools, tools for data restructuration.

Description of Documents with Ontologies

For increasing the efficiency in the processing an electronic document requires the presence of metadata that describe the structure and the semantics of the data. One of the possible approaches to describe the information placed in the document is the one based on ontologies. By ontology we understand the knowledge base of the special type that can be easily read, understood, alienate from the developer and/or divided physically by its users [4].

Ontology based approach has the following advantages:

- it is understandable for people;
- the user who develops the ontology do not need a special qualification;
- one document can be described with several ontologies.

An ontology based approach was chosen as the one for solving the task described above [1]. In this approach the ontology describes not only the structure but also the content of the document. According to the suggested approach the ontology is used for describing the semantics of the data in the document and its structure. Let us take into account the specific of the tasks solvable in this article and specify the definition of the ontology. We will consider the ontology as a specification of a certain subject area, which includes the dictionary with terms (concepts) from this area and a set of connections between the terms which describe how the terms are correlated between them in this particular area.

The following base types of the relations are used for constructing the hierarchy of the ontology concepts:

- “is_a”;
- “part_of”;
- “synonym_of”.

You should remember that the given types are the base types and do not depend on the ontology, but the user needs the possibility to add the new relations, which would take into account the specific of the subject area.

The ontology also includes two types of nodes. Nodes from the first group describe the structure of the document (for example, table, date, occupation and etc.). These nodes describe the common concepts which do not depend on the subject area. Nodes from the second group describe the specific concepts which belong to this particular document. We will call structured nodes those from the first group and semantic nodes those from the second group. In Fig. 1 structured nodes have the dark color, and semantic nodes have the light color.

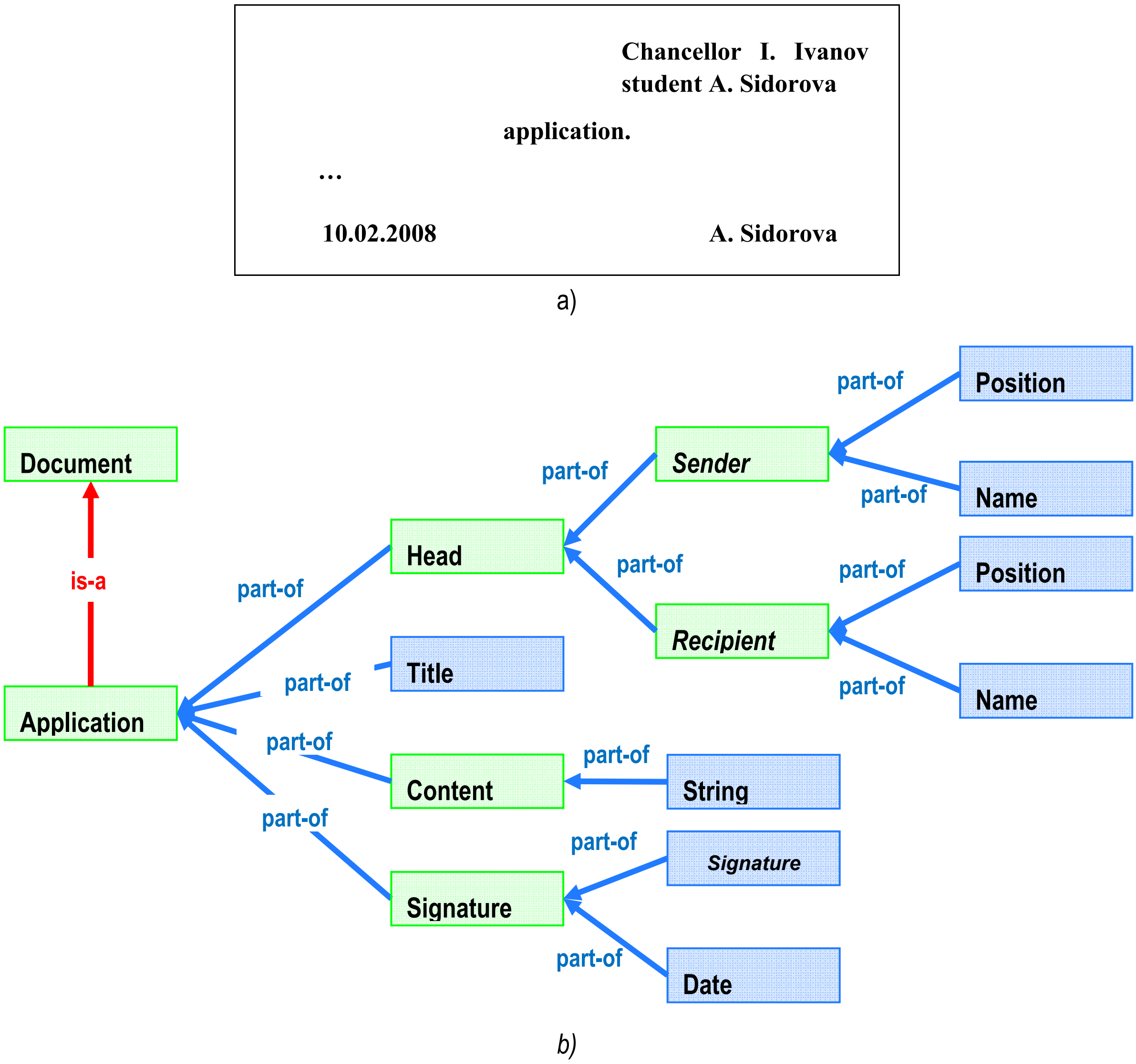
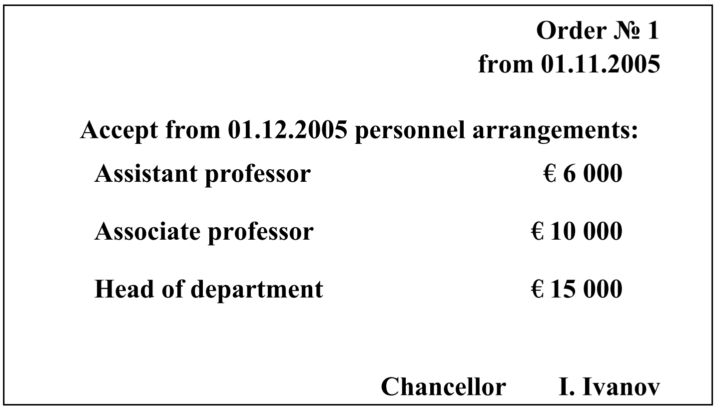


Fig. 1. Example of the document «Statement» (a) and ontology that describes the class of the documents «Statement» (b)

Actually in this context ontology is the hierarchical conceptual basis of the considered subject area. The ontology of the document is used for the document analysis, thanks to it we can obtain the required data because we know where to search and how to interpret the information.

If we present the document with the help of the ontologies then the task of comparison the ontology and existing document is reduced to the task to look for the concepts of the ontology in the document. As a consequence the system needs to answer the question: does this ontology describe the document or does not? The last question can be answered affirmatively if in the process of comparison all the ontology concepts were found in the document. It is necessary to search for the nodes which describe the structure of the document before we have a look for the nodes which contain the concepts of the document. Thus the original problem is reduced to the task to look for the common concepts in the text of the document on the basis of the formal descriptions.

In the given example (Fig. 2, b) all the nodes of the ontology are divided into two planes that is take into account under the comparison of the document (Fig. 2, a) and its ontology.

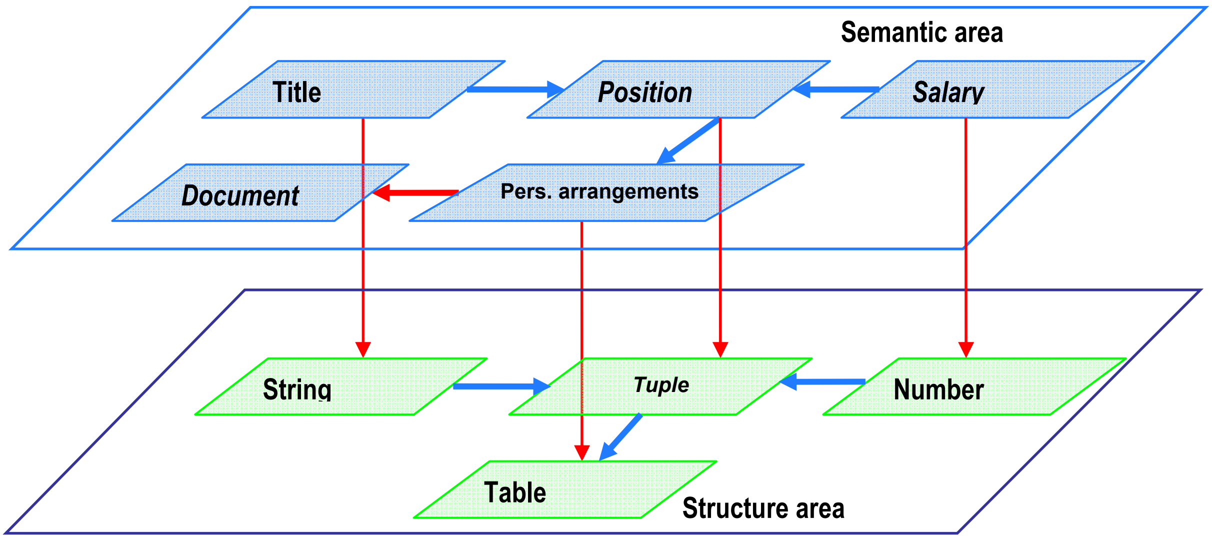

Order № 1
from 01.11.2005

Accept from 01.12.2005 personnel arrangements:

Assistant professor	**€ 6 000**
Associate professor	**€ 10 000**
Head of department	**€ 15 000**

Chancellor I. Ivanov

a)



b)

Fig. 2. Example of the document «Order» (a) and partitioning the ontology nodes of the document into two areas (b)

A repository of ontologies consists of 3 ontology levels (fig. 3). Ontologies which describe objects of specific system and take into account it features locate on the first level. Objects which are invariant to data domain locate on the second level. Objects of first level are described in terms of second level using relation «is_a» and «part_of». Objects of third level describe common conceptions and axioms which are used for describing objects of underlying levels. Third level and second level can be delivered to two part: describing of structures and describing of templates.

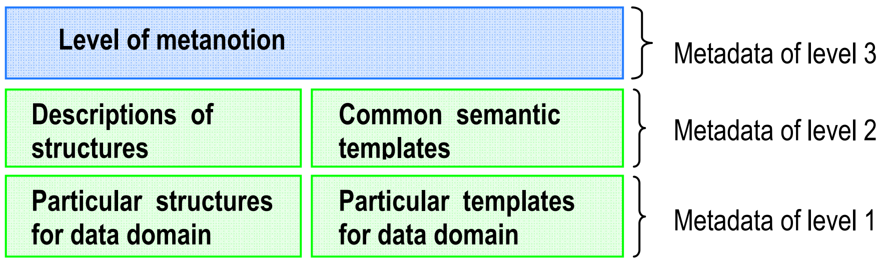


Fig. 3. Levels of metadata

Agent-Based Approach to Document Analysis

There are some requirements to the search process itself:

- the high rate of processing the large volumes of data;
- fault tolerance;
- scalability and
- adjustability to the users' need and changing conditions.

To solve the problem of extracting the common concepts base on formal descriptions we suggest an agent-based approach [2]. By the agent we understand a system that is aimed at specific goal, capable to interact with the environment and other agents to reach the goal [3]. This approach will satisfy all the requirements that were presented above to the search process, if during the system implementation all the advantages of multiagent systems will be realized.

With the use of this approach for each ontology node, that contains the common concept, the agent, which searches for this specific concept in the document, is created. The presence of the knowledge base (KB) is necessary condition to create an intelligent agent. Thus to determine the agents, that act in the system, we need to select the way of describing the KB, interaction with the environment and the way of collaboration with other agents. Tools to represent the KB are considered in the following section.

One of the most important properties of agents is a social significance or capability for interaction. As it was said earlier the agent is created for each ontology node that contains the common concept (semantic node). According to the classification of agents such agent is intentional.

The agent is aimed to solve two problems:

1. It divides into single components the entire list of the concepts patterns and starts simpler agents for search structure nodes.
2. It assembles all results obtained by the simpler agents.

Simpler agents mentioned above are reflector agents. They obtained the pattern and they aimed to find the fragments of the text that are fall under this pattern.

The communication between the agents is also an important question. All the mechanisms of communication can be divided into direct and indirect. A model of interaction called «contract network» is an example of direct communication. The mechanism of indirect communication is realized through the model «blackboard»:

– Contract network. This model assumes that all agents in the system can be divided into two classes: customers and contractors. The heart of the model is in solving the tasks by choosing for this the most appropriate agent. Customers are responsible for the distribution of tasks between the agents. Potential contractors analyze the requests provided by the customers, and if they can do it they send the request to the customer.
– Blackboard. This model is based on the model of the class board on which is presented the current state of the system where the agents are operating. All agents are continuously analyzing information on the board attempting to find a use to their possibilities. If at a certain moment the agent finds out the possibility to resolve the current task then it leaves a note on the board about the beginning of the work and it will place the results on the board after it completes the work.

Taking into account the special features of the task the combination of two models (both «contract network» and «blackboard») is implemented.

Both the architecture of the multiagent system and the process of document analysis are presented in Fig. 4.

Agent's Knowledge Base Presentation

One of the most important questions in the system is the one about presenting the KB of the agent. Nowadays presentation of the KB of the agent can be done in three different ways: with the use of ontologies, with the use of regular expressions and with the use of productions.

Presenting the knowledge of the agent with the help of the ontologies is the most expressive method which uses all the advantages of the explicit representation of the knowledge (Fig. 5). The advantage of this method is that we can use different ways to «proof» the node. For example it can be done through the simple coincidence of key phrase or request to the DB of the IS. Ontologies make it possible to describe different situations if it is impossible to find the exact coincidence. We can find more general or more define concept and so on.

The content of the analyzed document is presented in the form of special object model as the base of which was taken the object model of the document from Microsoft Word. The API functions were developed to give an access to this object model. The functions give an opportunity to operate with identical concepts when you work with documents in different formats. The API functions include functions to syntactic analysis of sentences, functions to evaluate different metrics between the concepts, functions to extract the information about the structure of the document. If additional operations are necessary to find the concept from the node they can be described in the script with the help of the API functions mentioned above. In the script you can also use the requests to the object model of the IS.

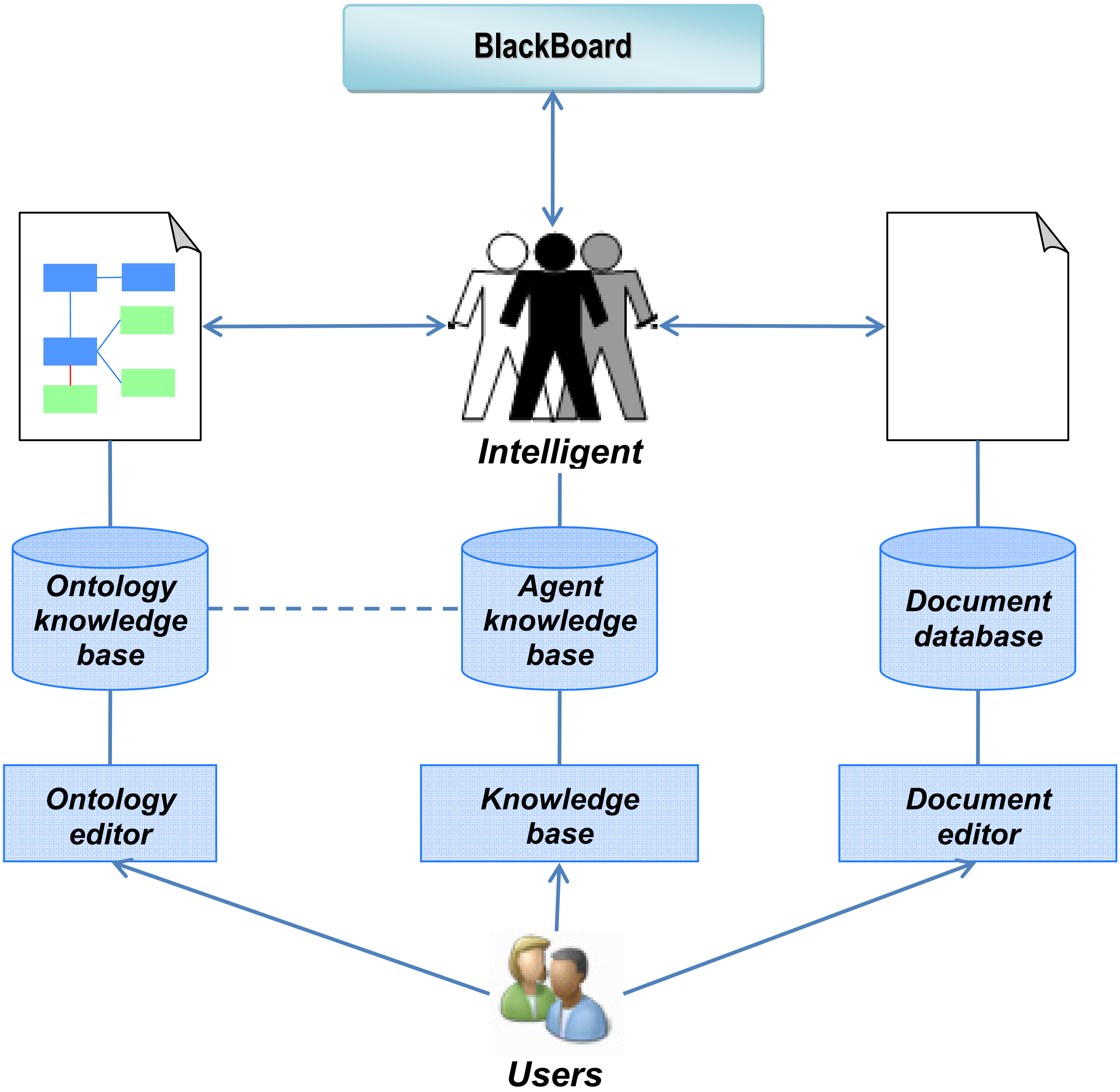


Fig. 4. Architecture of the SemanticDoc system

The second approach is the one that uses regular expressions. Expressions make it possible to consider various forms of the word and to work with large volumes of information [5]. It is necessary to remember that sometimes, especially for the users without special qualification, it is hard to construct a regular expression correctly. There is a special editor in the system to simplify the task. The editor allows working with the regular expressions in natural language. For example, the equivalent to the phrase «five-digit number» is «\d{5}» and etc. Furthermore the function of building a regular expression «upon the pattern» is also very useful. This means that on the examples given by the user the system can build the regular expression automatically. For example, the user typed two dates as examples «1.12.08» and «15.07.2006». The system will build an expression that is correspond with both given dates: «(\d{1,2}).(\d{1,2})((\d{4})|(\d{2}))».

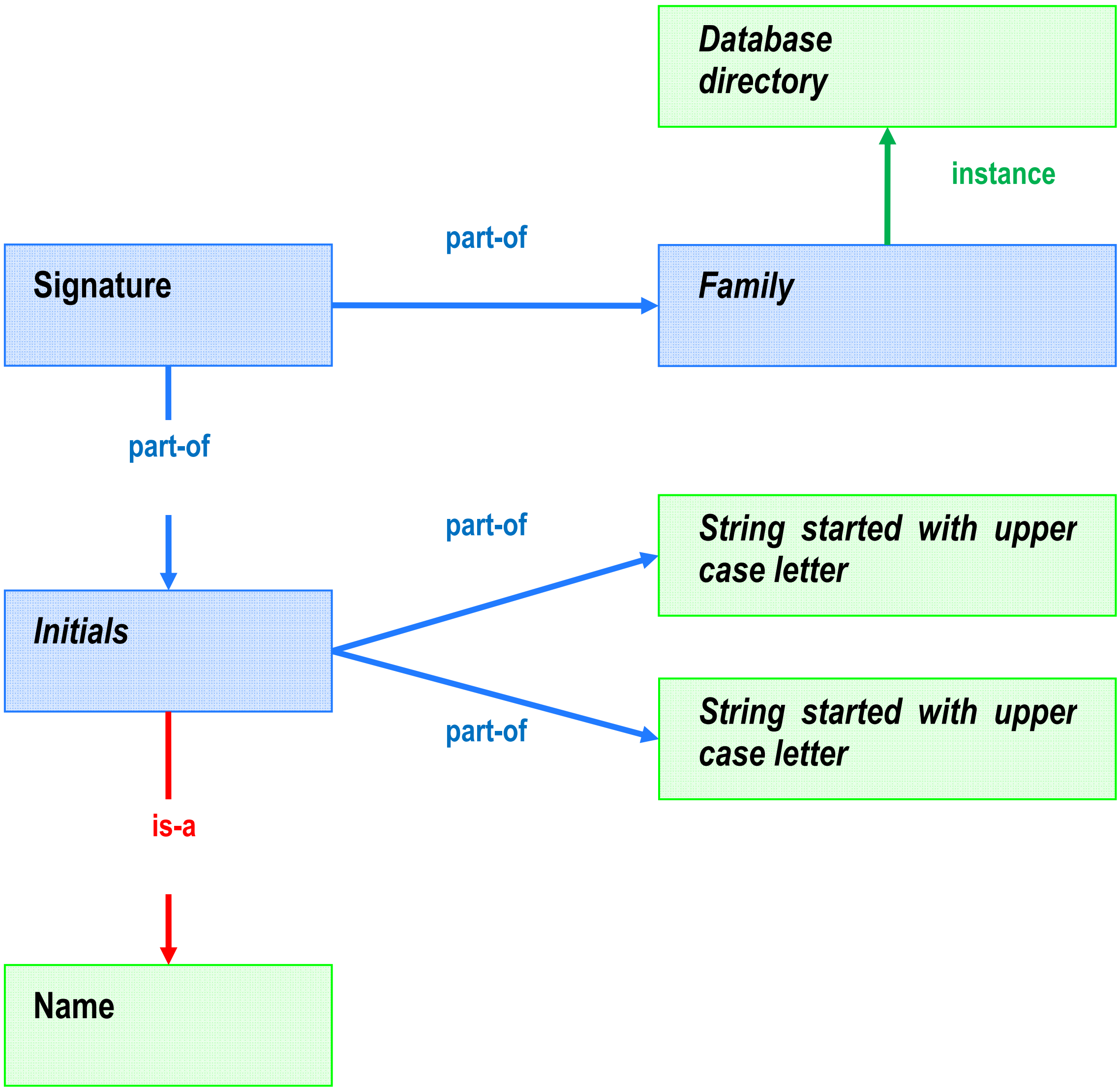


Fig. 5. Agent's knowledge base presentation with the help of the ontology

A drawback in the regular expressions is the fact that they do not take into account the location of the desired word or phrase during the search. It is possible to use both regular expressions and production rules, which are the third method of presenting the KB of the agent, to eliminate this defect.

Usually production rules are used to analyze the structure of the document. Special concepts can be used to set the conditions. For example, the rule is looking for the headings in the text can be formulated in the following way:

> ***If*** *(type of the paragraph differ from the type of previous and next paragraph)* ***and*** *(paragraph has center alignment),*
>
> ***then*** *paragraph is the heading.*

Conclusion

At the given moment the result of this work is the system SemanticDoc which was implemented on .NET. SemanticDoc is a multiagent system that matches the document and the ontology.

Two characteristics were brought in information retrieving to compare the quality of the results. The first is precision, the second is recall [6]. We can bring the similar characteristics for the system which matches the document and the ontology.

By precision (*P*) we will understand the number of correctly matched documents divided to the number of all matches done by the system. Under the recall (*R*) we will understand the number of correctly matched documents divided to the number of all existing matches.

Let the *N* be a number of all existing matches between the document and ontology, *M* – a number of matches that was done by the system, *A* – a number of correctly matched documents and ontologies. Then:

$$P = A/M \quad \text{and} \quad R = A/N .$$

Usually recall conflicts with precision and in practice it is impossible to reach both precision and recall.

The work on estimation this parameters was not conducted but the next stage will be evaluating the *P* and *R* after the experiments on the real documents.

The tools for document analysis can be used both for reduction the labor intensity of the users and for support the solution of subject area analysis done by the developers. In this case it is proposed the deep integration of the functional subsystem which includes not only the development tools, but also the tools for the end users. This given the possibility of designing the CASE-technology intended for the creation dynamically adjusted IS with the possibility to adapt to the changeable environment base on the feedback and smart document analysis.

Within this work the formal model of electronic document and ontology with regard to this task is developed, and base on it the existing object model of IS, metadata and algorithms to manage the documents become more specific.

So, listed above tasks can be solved by described approach:

- semantic indexing of documents;
- intelligent search;
- intelligent classification of documents;
- information extraction from not structured documents ;
- support of analyst work.

Acknowledgments

This work was supported by the Russian Foundation for Basic Research (Project 08-07-90006-Бел_а) and by the Russian Humanitarian Scientific Fund (Project 09-02-00373В/И).

Bibliography

[1] Ланин В.В. Интеллектуальное управление документами как основа технологии создания адаптируемых информационных систем // Труды международной научно-технической конференций «Интеллектуальные системы» (AIS'07). Т. 2 / М.: Физматлит, 2007. С. 334-339.

[2] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. М.: Эдиториал, УРСС, 2002.

[3] Рассел С. Искусственный интеллект: современный подход. М.: Издательский дом «Вильямс», 2006.

[4] Хорошевский В.Ф., Гаврилова Т.А. Базы знаний интеллектуальных систем. СПб.: Питер, 2001.

[5] Фридл Дж. Регулярные выражения. СПб.: Питер, 2003.

[6] Weal M.J., Kim S., Lewis P.H., Millard D.E., Sinclair P.A.S., De Roure D.C., Nigel R. Ontologies as facilitators for repurposing web documents / Shadbolt. Southampton, 2007.

Author's Information

Vyachslav Lanin *– Perm State University, Postgraduate student of Computer Science Department; 15, Bukirev st., Perm, Russia; e-mail: lanin@psu.ru.*

Elena Mozzherina *– Perm State University, Candidate for a master's degree of Computer Science Department; 15, Bukirev st., Perm, Russia; e-mail: mozzherina@gmail.com.*

ON THE EXTERIOR PENALTY FUNCTION METHOD FOR THE CONSTRAINED OPTIMAL CONTROL PROBLEM FOR QUASILINEAR PARABOLIC EQUATIONS

Mahmoud Farag

Abstract: *This paper presents the numerical solution of a constrained optimal control problem (COCP) for quasilinear parabolic equations. The COCP Is converted to unconstrained optimization problem (UOCP) by applying the exterior penalty function method. The numerical algorithm for solving UOCP using the conjugate gradient method (CGM) is given. The computing optimal controls are helped to identify the unknown coefficients of the quasilinear parabolic equation. Numerical results are reported.*

Keywords: *optimal control, quasilinear parabolic equations, penalty function method, finite difference methods.*

ACM Classification Keywords: *G.1 Numerical Analysis, G.1.6 Optimization, G.1.8 Partial Differential Equations.*

Nomenclature and Notations

E_N	The N-dimensional Euclidean space
D	a bounded domain of E_N
Ω	$\{(x,t): x \in D,\ t \in (0,T)\}$
$\phi(x) \in L_2(D), g_0(t), g_1(t) \in L_2(0,T), f_0(t), f_1(t) \in L_2(0,T)$	Given functions
r_1 , r_2 , l , T , $\omega = (w_1, w_2, \ldots, w_N) \in E_N$	Given numbers
$\alpha \geq 0,\ \beta_0 \geq 0,\ \beta_1 \geq 0,\ \beta_0 + \beta \geq 0$ $\zeta_0,\ \mu_0,\ \zeta_1,\ \mu_1 > 0,\ R > 0$	Given numbers
$U = \{u : (u_1, u_2, \ldots, u_N) \in E_N, \|u\|_{E_N} \leq R\}$	Control space
C	Constant not depending on δu
A_m	Positive numbers, $\lim_{m \to \infty} A_m = +\infty$

$L_2(D)$	Banach space which consisting of all measurable functions on D with the norm $\Vert y \Vert_{L_2(D)} = \sqrt{\int_D \vert y \vert^2 \, dx}$
$L_2(\Omega)$	Hilbert space which consisting of all measurable functions on Ω with $\langle y_1, y_2 \rangle_{L_2(\Omega)} = \int_0^l \int_0^T y_1(x,t) \; y_2(x,t) \; dx \;\; d$ $\Vert y \Vert_{L_2(\Omega)} = \sqrt{\langle y, y \rangle_{L_2(\Omega)}}$
$L_2(0,l)$	Hilbert space which consisting of all measurable functions on $(0,l)$ with $\langle y_1, y_2 \rangle_{L_2(0,l)} = \int_0^l y_1(x) \; y_2(x) \; dx \;\; , \;\; \Vert y \Vert_{L_2(0,l)} = \sqrt{\langle y, y \rangle_{L_2(0,l)}}$
$W_2^{1,0}(\Omega)$	$W_2^{1,0}(\Omega) = \left\{ y \in L_2(\Omega) \;\; and \;\; \frac{\partial y}{\partial x} \in L_2(\Omega) \right\}$ is a Hilbert space with $\langle y, y \rangle_{W_2^{1,0}(\Omega)} = \sqrt{\Vert y \Vert^2_{L_2(\Omega)} + \left\Vert \frac{\partial y}{\partial x} \right\Vert^2_{L_2(\Omega)}}$ $\Vert y \Vert_{W_2^{1,0}(\Omega)} = \int_\Omega [\, y_1 \;\; y_2 \; + \; \frac{\partial y_1}{\partial x} \frac{\partial y_2}{\partial x}] \; dx \;\; dt$
$W_2^{1,1}(\Omega)$	$W_2^{1,1}(\Omega) = \left\{ y \in L_2(\Omega) \;\; and \;\; \frac{\partial y}{\partial x}, \frac{\partial y}{\partial t} \in L_2(\Omega) \right\}$ is a Hilbert space with $\langle y, y \rangle_{W_2^{1,1}(\Omega)} = \sqrt{\Vert y \Vert^2_{L_2(\Omega)} + \left\Vert \frac{\partial y}{\partial x} \right\Vert^2_{L_2(\Omega)} + \left\Vert \frac{\partial y}{\partial t} \right\Vert^2_{L_2(\Omega)}}$ $\Vert y \Vert_{W_2^{1,1}(\Omega)} = \int_\Omega [\, y_1 \;\; y_2 \; + \; \frac{\partial y_1}{\partial x} \frac{\partial y_2}{\partial x} + \frac{\partial y_1}{\partial t} \frac{\partial y_2}{\partial t}] \; dx \;\; dt$
$V_2^{1,0}(\Omega)$	Banach space consisting of elements the space $W_2^{1,0}(\Omega)$ with the norm $\Vert y \Vert_{V_2(\Omega)} = vrai \max_{0 \le t \le T} \Vert y(x,t) \Vert_{L_2(D)} + \sqrt{\int_\Omega \left\vert \frac{\partial y}{\partial t} \right\vert^2 dx}$
$V_2^{1,0}(\Omega)$	Subspace of $V_2(\Omega)$, the elements of which have in sections $D_t = \{(x,\tau) : x \in D, \tau = T\}$ traces from all $t \in [0,T]$, continuously charging from $t \in [0,T]$ in the norm $L_2(D)$.

Introduction

Optimal control problems for partial differential equations are currently of much interest. A large amount of the theoretical concept which governed by quasilinear parabolic equations has been investigated in the field of optimal control problems [Belmiloudi, 2004], [Ryu, 2004]. These problems have dealt with the processes of hydro- and gasdynamics, heat physics, filtration, the physics of plasma and others [Farag, 2004], [Iskenderov, 1974]. From the mathematical point of view, the definition and refinement of the unknown parameters of the model present the problem of identification and optimal control of partial differential equations. The importance of investigating the identification and optimal control problems was developed in [Lions, 1973], [Farag, 2003].

This paper presents the numerical solution of a constrained optimal control problem (COCP) for quasilinear parabolic equations. The COCP Is converted to unconstrained optimization problem (UOCP) by applying the exterior penalty function method [Xing,1994]. The numerical algorithm for solving UOCP using the conjugate gradient method is given [Damean, 2000]. The computing optimal controls are helped to identify the unknown coefficients of the quasilinear parabolic equation. Numerical results are reported.

The heat exchange process described by a partial differential equation of quasilinear parabolic type as follows:

$$\frac{\partial y}{\partial t}-\frac{\partial}{\partial x}\left(\lambda\,(y,u)\,\frac{\partial y}{\partial x}\right)+Z\,\,(y,u)\,\,y=F(x,t)\,,(x,t)\in\Omega \tag{1}$$

with the initial condition and the boundary conditions

$$y\,(x,0)=\phi\,(x)\quad,\;\;x\,\in\,D \tag{2}$$

$$\lambda\,(y,u)\left.\frac{\partial y}{\partial x}\right|_{x=0}=g_0\,(t)\;\;,\;\;\lambda\,(y,u)\left.\frac{\partial y}{\partial x}\right|_{x=l}=g_1\,(t)\,,\;0\le\,t\,\le\,T \tag{3}$$

On the set U under the conditions (1)-(3) and additional restrictions

$$\zeta_0\,\le\,\lambda\,(y,u)\le\,\mu_0\,,\;\zeta_1\,\le\,Z(y,u)\,\le\mu_1\,,\;r_1\,\le\,\,y(x,t)\le\,r_2 \tag{4}$$

is required to minimize the function

$$J_\alpha(u)=\,\beta_0\,\int_0^T\,\left[y\,(0,t)-f_0(t)\right]^2dt+\beta_1\int_0^T\,\left[y\,(l,t)-f_1\,(t)\right]^2\,\,dt+\alpha\,\|u-\omega\|_{E_N}^2 \tag{5}$$

The solution of the reduced problem (1)-(3) explicitly depends on the control u . Therefore, we shall also use the notation $y=y\,(x,t;u)$. Based on adopted assumptions and the results of [Ladyzhenskaya, 1973] follows that for every $u\in U$ the solution of the problem (1)-(4) is existed, unique and $|y_x|\le C,\,\forall\,(x,t)\in\Omega,\,\forall\,u\in U$.

Optimal control problems of the coefficients of differential equations do not always have solution [Goebel, 1979]. In [Farag, 2003], we proved the existence and uniqueness of the solution of problem (1)-(5) as follows:

Lemma (1)

At above adopted assumptions for the solution of the reduced problem (1)-(5) the following estimation is valid

$$\| \delta y \|_{V_2^{1,0}(\Omega)} \le C \sqrt{ \left\| \lambda(y,u) \frac{\partial y}{\partial x} \right\|^2_{L_2(\Omega)} + \| \delta Z(y,u)\, y \|^2_{L_2(\Omega)} } \quad (6)$$

Lemma (2): The function $J_0(u)$ is continuous on U.

Theorem (1) : The problem (1)-(5) at any $\alpha \ge 0$ has at least one solution.

Theorem (2): The problem (1)-(5) at any $\alpha > 0$, at almost all $\omega \in E_N$ has a unique solution.

UOCP and CBVP

The inequality-constrained problem (1) through (5) is converted to a problem without inequality constraints by adding a penalty function to the objective (5), yielding the minimizing following function:

$$Y_{\alpha,m}(u) = f_\alpha(u) + A_m \int_0^l \int_0^T [\, B_1(y,u) + B_2(y,u) + P_1(y) + P_2(y)]\, dx\, dt \quad (7)$$

under the condition (1)-(3), where

$$P_1(y) = [\max\{r_1 - y(x,t;u);0\}]^2 \qquad B_1(y,u) = [\max\{\zeta_0 - \lambda(y,u);0\}]^2 + [\max\{\lambda(y,u) - \mu_0;0\}]^2$$

$$P_2(y) = [\max\{y(x,t;u) - r_2;0\}]^2 \qquad B_2(y,u) = [\max\{\zeta_1 - \lambda(y,u);0\}]^2 + [\max\{\lambda(y,u) - \mu_1;0\}]^2$$

The problem (7) and (1)-(3) is called **UOCP**. It is assumed that the following conditions are fulfilled:

a) The functions satisfy the Lipshitz condition for u .

b) The first derivatives of the functions $\lambda(y,u)$, $Z(y,u)$ with respect to u are continuous functions.

c) For any $u \in U$ *s.t.* $\|u\|_{E_N} \le R$ the functions $\frac{\partial \lambda(y,u)}{\partial u}, \frac{\partial Z(y,u)}{\partial u}$ belong to $L_\infty(\Omega)$.

d) $\int_0^l \int_0^T \frac{\partial \lambda(y,u)}{\partial u}\, dx\, dt$, $\int_0^l \int_0^T \frac{\partial Z(y,u)}{\partial u}\, dx\, dt$ are bounded in E_N

Theorem (3): It is assumed that the above conditions are satisfied. The function $\Psi(x,t) \in W_2^{1,1}(\Omega)$ is a solution in of the following conjugate boundary value problem (**CBVP**) [Vassiliev, 1980] :

$$\frac{\partial \Psi}{\partial t}+\frac{\partial}{\partial x}\left(\lambda(y,u)\frac{\partial \Psi}{\partial x}\right)-\frac{\partial \lambda(y,u)}{\partial y}\frac{\partial \Psi}{\partial x}\frac{\partial y}{\partial x}-\left(\frac{\partial Z(y,u)}{\partial y}y+Z(y,u)\right)\Psi$$
$$=A_m\left[\frac{\partial B_1(y,u)}{\partial y}+\frac{\partial B_2(y,u)}{\partial y}+\frac{\partial P_1(y)}{\partial y}+\frac{\partial P_2(y)}{\partial y}\right],(x,t)\in\Omega \tag{8}$$

$$\Psi(x,T)=0 \ , \ x \in D \tag{9}$$

$$\lambda(y,u)\frac{\partial \Psi}{\partial x}=\begin{cases} 2\beta_0\,[\,y(x,t)-f_0(t)\,] & x=0 \\ -2\beta_1\,[\,y(x,t)-f_1(t)\,] & x=l \end{cases} \ , \ 0\le t\le T \tag{10}$$

where y is the solution of problem (1)-(3) for $u\in U$.

For the sufficient differentiability conditions of the function $\Upsilon_{\alpha,m}(u)$, we have the following theorem:

Theorem (4): It is assumed that the above conditions are satisfied. The function $\Upsilon_{\alpha,m}(u)$ is Frechet differentiable and its gradient satisfies the equality

$$\frac{\partial \Upsilon_{\alpha,m}}{\partial u}=-\frac{\partial H}{\partial u}\equiv\left(-\frac{\partial H}{\partial u_1},-\frac{\partial H}{\partial u_2},\dots,-\frac{\partial H}{\partial u_N}\right) \tag{11}$$

where $H(y,\Psi,u)$ is the Hamiltonian function defining as follows

$$H(y,\Psi,u)\equiv-\int_\Omega[\lambda(y,u)\frac{\partial \Psi}{\partial x}\frac{\partial y}{\partial x}+Z(y,u)\,y\,\Psi$$
$$+A_m\{B_1(y,u)+B_2(y,u)\}\,]\,dx\,dt-\alpha\,\|u-\omega\|^2_{E_N} \tag{12}$$

The Iterative Algorithm for Solving UOCP

The following iterative algorithm is developed (k being iteration numbers). In view of relations (1)-(7) one considers the following iterative algorithm:

Step 0: Choose $k=1$, $u_{Nc}\in V$, $\omega_{Nc}\in E_N$, $\varepsilon_1>0$, $\varepsilon_2>0$

Step 1: Compute $Y^{(k)}(x,t)$, that is, the state system described by equations (1)-(4):

$$\frac{\partial y^{(k)}}{\partial t}-\frac{\partial}{\partial x}\left(\lambda(y^{(k)},u^{(k)})\frac{\partial y^{(k)}}{\partial x}\right)+Z(y^{(k)},u^{(k)})\,y^{(k)}=F(x,t),(x,t)\in\Omega$$

$$y^{(k)}(x,0)=\phi(x)\ ,\ x\in D$$

$$\lambda(y^{(k)},u^{(k)})\frac{\partial y^{(k)}}{\partial x}\bigg|_{x=0}=g_0(t),\ \lambda(y^{(k)},u^{(k)})\frac{\partial y^{(k)}}{\partial x}\bigg|_{x=l}=g_1(t)\ ,\ 0\le t\le T$$

Step 2: Compute $\Psi^{(k)}(x,t)$, that is, the adjoint state system described by equations (10)-(14)

$$\frac{\partial\Psi^{(k)}}{\partial t}+\frac{\partial}{\partial x}\left(\lambda(y^{(k)},u^{(k)})\frac{\partial\Psi^{(k)}}{\partial x}\right)-\frac{\partial\lambda(y^{(k)},u^{(k)})}{\partial y^{(k)}}\frac{\partial\Psi^{(k)}}{\partial x}\frac{\partial y^{(k)}}{\partial x}$$

$$-\left(\frac{\partial Z(y^{(k)},u^{(k)})}{\partial y^{(k)}}y^{(k)}+Z(y^{(k)},u^{(k)})\right)\Psi^{(k)}$$

$$=A_m\left[\frac{\partial B_1(y^{(k)},u^{(k)})}{\partial y^{(k)}}+\frac{\partial B_2(y^{(k)},u^{(k)})}{\partial y^{(k)}}+\frac{\partial P_1(y^{(k)})}{\partial y^{(k)}}+\frac{\partial P_2(y^{(k)})}{\partial y^{(k)}}\right],(x,t)\in\Omega$$

$$\Psi^{(k)}(x,T)=0\ ,\ x\in D$$

$$\lambda(y^{(k)},u^{(k)})\frac{\partial\Psi^{(k)}}{\partial x}=\begin{cases}2\beta_0\,[\,y^{(k)}(x,t)-f_0(t)\,] & x=0\\ -2\beta_1\,[\,y^{(k)}(x,t)-f_1(t)\,] & x=l\end{cases}\ ,\quad 0\le t\le T$$

Step 3: Find optimal control $u^{*(k)}$ using conjugate gradient method.

Step 4: Compute $\lambda^*_{Nc/2}=\lambda(y^{(k)},u^{*(k)}_{Nc/2})\ ,\ Z^*_{Nc/2}=Z(y^{(k)},u^{*(k)}_{Nc/2})$

Step 5: If $\left|\lambda_{Exact}-\lambda^*_{Nc/2}\right|<\varepsilon_1\ ,\left|Z_{Exact}-Z^*_{Nc/2}\right|<\varepsilon_2$ then terminate the procedure, otherwise set $Nc=Nc+2\ ,\ k=k+1$ and go to step 1.

Numerical Results

The problem (1)-(7) is considered as one of the identification problems on definition of unknown coefficients of parabolic quasilinear equation type. The numerical results were carried out for the following example of exact solution, input data:

$$y=x+t\ ,\lambda(y,u)=\tan^{-1}(y),Z(y,u)=y^2(1-y^2)\ ,0<x<0.9\ ,0<t<0.01$$

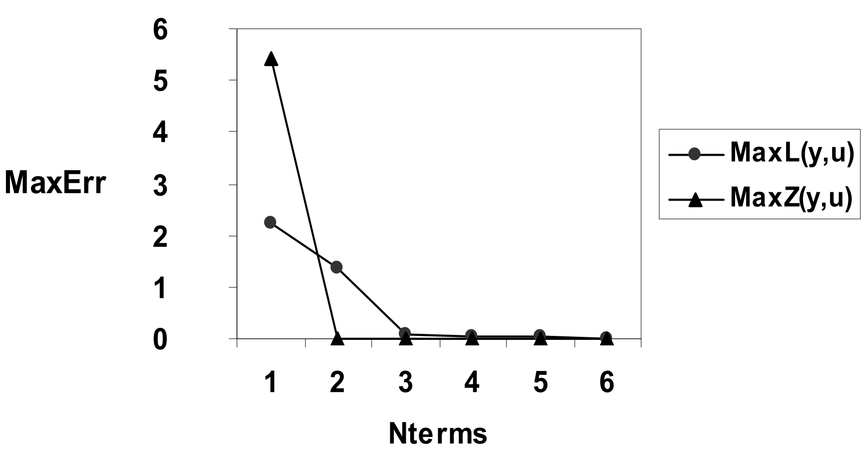
$$f_0=g_0=\tan^{-1}x\ ,f_1=g_1=\tan^{-1}(0.9+t)\ ,\ \phi(x)=x,F(x,t)=\frac{y}{1+y^2}+\frac{y^3}{1-y^2}$$

In Fig. 1, the curves denoted by MaxL(y,u) and MaxZ(y,u) are the maximum absolute errors

$$MaxL\ (y,u) = \max\ \left|\lambda_{Exact} - \lambda^{*}_{Nc/2}\right|$$
$$MaxZ\ (y,u) = \max\ \left|Z_{Exact} - Z^{*}_{Nc/2}\right|$$

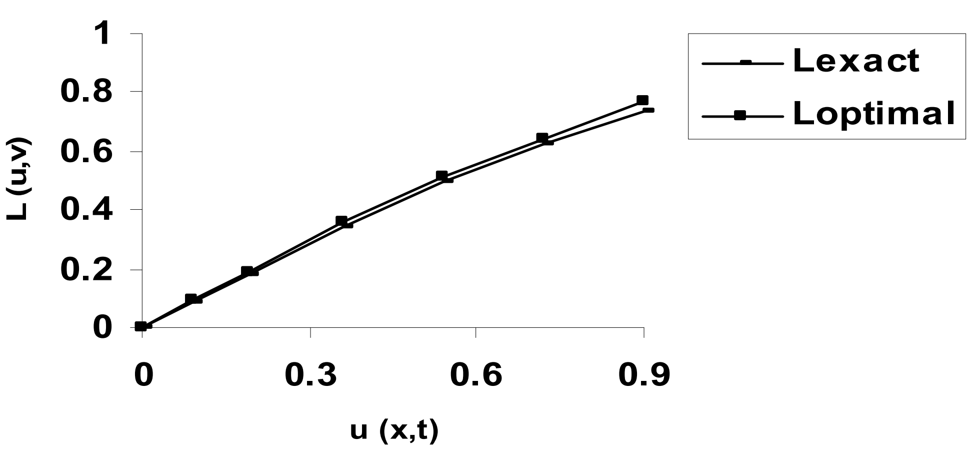
It is clear the maximum absolute errors decrease as $Nc \equiv Nterms$ increase.

Fig. 1 : Number of terms versus Max Absolute Error



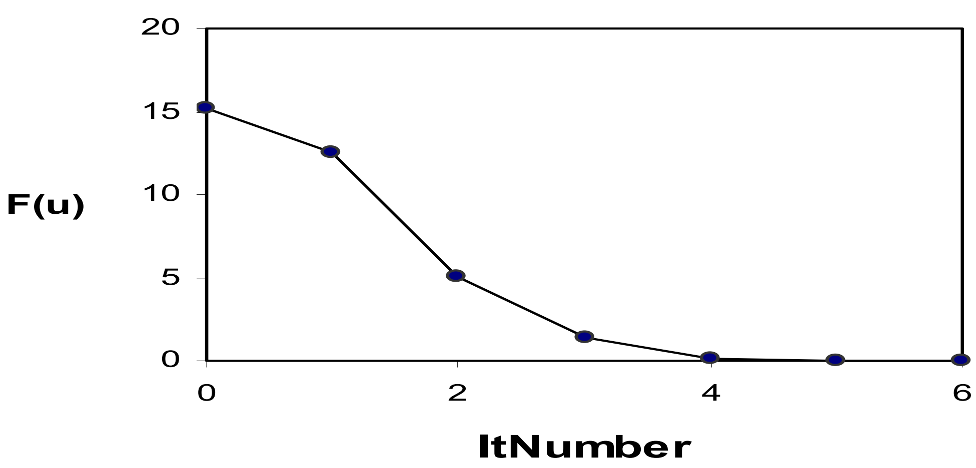
In Fig. 2, the curves denoted by Lexact and Loptimal $(\lambda_{Exact}\ ,\ \lambda_{optimal}\ )$ are the exact values and approximate values with the optimal control $\overset{*}{u}$.
By increasing Nc, $\lambda(y,u)$ will agree with the exact value.

Fig. 2: Coefficients of QPDEs (Exact & Optimal)



The values of $\Upsilon_{\alpha,m}(u) = F(u)$ versus the iteration numbers are displayed in Fig. 3.

Fig. 3 : Iteration Number versus F(u)



Bibliography

[Belmiloudi, 2004] A. Belmiloudi. On some control problems for nonlinear parabolic equations. Inter. J. of Pure and Appl. Math., v. 11, No. 2, 115-169, 2004.

[Damean, 2000] N. Damean. New algorithm for extreme temperature measurements. Adv. In Eng. Soft. 31, 275-285, 2000.

[Farag, 2004] M. H. Farag. A stability theorem for constrained optimal control problems. J. Comput. Math. 22, No.5, 633-640, 2004.

[Farag, 2003] M. H. Farag. Necessary optimality conditions for constrained optimal control problems governed by parabolic equations. J. Vib. Control 9, No.8, 949-963, 2003.

[Farag, 2003] M. H. Farag. On the derivation of discrete conjugate boundary value problem for an optimal control parabolic problem. N. Z. J. Math. 32, No.1, 21-31, 2003.

[Goebel, 1979] M. Goebel. On existence of optimal control. Math. Nuchr. 93, 67-73, 1979.

[Iskenderov, 1974] D. Iskenderov. On a certain inverse problem for quasilinear parabolic equations. Diff. Urav., 10(5), 890-895, 1974.

[Ladyzhenskaya, 1973] A. Ladyzhenskaya. Boundary value problems of mathematical physics. Nauka, Moscow, Russian, 1973.

[Lions, 1973] J.-L Lions. Optimal control in distributed systems. Uspikian Math. Nauka, 28(4), 1973.

[Vassiliev, 1980] F. Vassiliev. Numerical Methods for solving external problems. Nauka, Moscow, Russian, 1980.

[Ryu, 2004] S.-U. Ryu. Optimal control problems governed by some semilinear parabolic equations. Nonlinear analysis, v. 56, iss. 2, 241-252, 2004.

[Xing,1994] A-Q. Xing. The exact penalty function method in constrained optimal control problems. J. Math. Anl. and Applics., 186, 514-522, 1994.

Author's Information

M. H. Farag *– Department of Mathematics and Computer Science, Faculty of Education, Al-Rustaq, Sultanate of Oman, e-mail: farag5358@yahoo.com*

DISTRIBUTED VIRTUAL LABORATORIES FOR SMART SENSOR SYSTEM DESIGN

Oleksandr Palagin, Volodymyr Romanov, Vitalii Velychko, Igor Galelyuka, Volodymyr Fedak, Volodymyr Grusha, Dmytro Artemenko, Oksana Galelyuka

Abstract: *In the article it is considered preconditions and main principles of creation of virtual laboratories for computer-aided design, as tools for interdisciplinary researches. An important feature of this project is using the advanced multi-dimensional access method for organizing the information base of the Virtual laboratory. Virtual laboratory, what are offered, is worth to be used on the stage of the requirements specification or EFT-stage, because it gives the possibility of fast estimating of the project realization, certain characteristics and, as a result, expected benefit of its applications. Using of these technologies already increase automation level of design stages of new devices for different purposes. Proposed computer technology gives possibility to specialists from such scientific fields, as chemistry, biology, biochemistry, physics etc, to check possibility of device creating on the basis of developed sensors. It lets to reduce terms and costs of designing of computer devices and systems on the early stages of designing, for example on the stage of requirements specification or EFT-stage.*

An instance of using the VLCAD is designing the Portable Device "Floratest" as Tool for Estimating of Megalopolis Ecology State. Portable device "Floratest" is aimed for express-diagnostic of plant state. It is developed in the V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine. Party of this device is manufactured and transferred to organizations, worked in the agricultural sector, environmental protection area, mineral fertilizer production etc. for working out of methodical tools. Using of the device for estimating of megalopolis ecology state by means of evaluation of green plant state is described in the article. Together with Megalopolis Ecomonitoring and Biodiversity Research Center of National Academy of Sciences of Ukraine there were got results of experimental researches of influence detecting of heavy metals and harmful substances on the trees and plants in Kiev.

Keywords: *Virtual Laboratory; Computer-Aided Design; Access Methods; Distributed System, Kautsky effect, chlorophyll, chlorophyll fluorescence induction, fluorescence, fluorometer, portable device, ecology.*

ACM Classification Keywords: *J.6 Computer-Aided Engineering –Computer-Aided Design (CAD); D.4.3 File Systems Management – Access Methods; K.4.3 Organizational Impacts – Computer-Supported Collaborative Work; J.3 Life and Medical Sciences - Biology and Genetics, J.7 Computers in Other Systems – Real Time, C.3 Special-Purpose and Application-Based Systems – Microprocessor/microcomputer applications*

Introduction

Fast spreading of market relations and competition between manufacturers of different (including scientific) production and information services makes very actual the acceleration of development of theory and methods of computer-aided design of computer devices and biosensors. Actual design of devices and systems, which is often used, needs a lot of time, material, and human resources. If one needs to make a small set of devices by means of actual design, the price of final production becomes very high. Therefore, manufactures of computer devices get very complicated issue, which consists in time and price reduction of new devices design. Only after solving of this issue the new devices of own design will be able to become competitive on domestic and world markets.

To minimize these design expenses to reach high level of competitive recently side by side with actual design it is begun to use a virtual design. These methods realized by means of virtual laboratories of computer-aided design (VLCAD), which are based on advanced access methods and worth to be used on the stage of the requirements specification or EFT-stage, because it gives the possibility of fast estimating of the project realization, certain characteristics and, as a result, expected benefit of its applications.

Market analysis and joint discussion confirm the acute necessity in the developing of new virtual design methods and in the creating on their base open VLCAD, main feature of which is possibility to use such remote laboratory by specialists in different science branches, without education in information technologies and instrumentation.

Preconditions and Main Principles of Virtual Laboratory Creation

One of problems, which are met by developers of new devices for different fields of science and engineering, is existence of more than 15 thousands of such fields or disciplines to date. Naturally to carry out researches or create a new device developers must have knowledge from disciplines, which refer to developed device. Therefore it is important to orientate new computer technology for interdisciplinary researches, which occur on boundary of several science fields or disciplines.

Urgency of these researches is caused by absence of computer technology of smart devices designing for interdisciplinary researches in Ukraine and Bulgaria. It does not allow to test on computer models the performance of designed devices, which are created on the base of new effects or sensors. To date to develop new device or to check the possibility of its creations and operation it is necessary to invite specialists in information technology, electronics and circuit technology on the commercial base. Getting results in such way is very expensive and, as usual, is not supported with necessary funds. This again confirms acute necessity of design technology development and creating on their base the open virtual laboratories, the main feature of which is possibility to use these virtual laboratories by specialists from different science fields, especially non-specialists in the field of information technology and instrument making.

Good solution of this problem is to create on the base of information technologies the special hardware-software tools [Palagin and Sergiyenko , 2003], which in convenient mode (for example, with help of dialogues) allows sensor developer to check possibility of creating of new devices and the device model. Such tool has to give possibility to create a model set of certain device (e.g. functional, electrical, operational etc.), including prior parameters calculations, project of circuit board and set of design documentations (e.g. cost, performance, validity, size, reliability etc.). Description of sensor or its model should be incoming data for such design system.

Now on the world market there are a lot of software for computer-aided design (CAD), which allow to automotive design of new devices and systems and analyze them in different ways [Gavrilov, 2000]. But for skilled usage of such CAD software it is necessary to have special skills in circuit technology, electronics and instrument engineering, and also know this CAD software perfectly. It is clear, that sensor developers, who are mainly chemists, biologists, biochemists, physicists etc, have no enough possibility and skills to use such complicated CAD software for designing of new devices on basis of developed sensors. In such case they need help of CAD specialists. But it is very expensive service. Therefore in most cases sensor developer leave sensor "in quiet" and switch his attention to another tasks.

It is necessary to note, that only by paying attention to the design process of computer devices it will be possible to reach a high level of competitiveness of scientific developments, what lets in the future to take up notable place on the world market. It is easily to see, that most devices have the same structure, to be exact, they consist of sensor, measuring channel, data processor, interface and additional subsystems. That's why process of designing could be easily formalized.

To solve this problem within the bounds of international Ukrainian-Bulgarian project it is began developing of virtual laboratory for computer-aided design for computer device designing [Palagin et al, 2007]. The VLCAD is being created on the virtual methods of design [Galelyuka, 2008]. Offered virtual laboratory are created on the base of formalized representation of theoretic knowledge, principles of organization, methods and facilities of computer-aided design and testing information-measuring systems and devices, in particular on the base of subject field ontology. For VLCAD creating it is used the methodology of system integration [Palagin and Kurgaev, 2003] concerning base methods and tools, on which it is created. In the methodology basis it is putted system approach to tasks of analysis and synthesis of both VLCAD component and object of designing, and, first of all, forming knowledge system of interdisciplinary nature and its computer ontology. Proposed VLCAD is open system.

Mentioned VLCAD allows sensor developer to:

- check possibility of creating of devices and computer facilities (including portable devices) on basis of developed sensors without involving specialists in circuit technology and instrument engineering at the stage of EFT-project. It allows reducing terms and costs on this stage;
- avoid expensive actual tests on the stage of device creating by replacing with virtual methods of designing and testing;
- prepare set of design documentations on designed device in the automotive mode without involving corresponding specialists. Next stage is to send design documentations to contract production for creating of test party of devices.

Terms "Virtual laboratory" and "Virtual design" appear lately, so, as usual, they are absent almost in all dictionaries. The word "Virtual" appeared in word literature a long time ago. "Virtuality" has almost all features of empirical reality with the exception of its direct presence. So, it is "reality, which is absence" or "present absence". Also, "virtual" is one, which has no physical embodiment. "Virtual reality" is comprehended as a part of reality, which is modeled by computer device. Since any laboratory is a part of reality, so taking into account above-stated, there can be formulated next term of "virtual laboratory": virtual laboratory is imagined laboratory, which has all features of real laboratory and is modeled by means of software and hardware.

In general, virtual laboratory is some information environment, which lets to conduct researches in the case, when there is no direct access to test subject. Researches can be conducted by means of mathematical models and with using of remote access to test object.

Somebody may work with physical objects in two ways:

- emulation of physical objects with defined level of approximation to reality;
- remote access to physical objects with defined capabilities of interacting.

The first method lets to get completely virtual analog of some environment, what is very practical. Disadvantage of this method is complexity of model creating, which is very approximate to reality.

The second method provides maximal approximating to reality. But it requires creating and supporting of remote access to test objects, but the number of access channels is limited. Server of laboratory setup, besides access to equipments, is able to give background and methodological materials to researcher. Remote experiment in most cases is conducted in such way. Researcher communicates laboratory setup server and send data for experiment. Server software conducts experiment and sends results as tables, graphics to researcher.

For realization of VLCAD it is decided to use the first method. But the second method is not set aside and in future it will be probably used as additional tool.

Virtual laboratories, in which experiments are conducting by means of mathematical models, differ from previous one by using mathematical or other model instead of real test object. These laboratories have no laboratory setup.

Creating of VLCAD

Before VLCAD creating, first of all, it is necessary to determine features of VLCAD as tool for interdisciplinary researches and what functions it has to have.

In general, VLCAD is a system for computer-aided design, but with certain difference. This difference is that for using any CAD system it is necessary to have deep knowledge in this software, instrument engineering, circuit technology and electronics. It is expected, that for using VLCAD users need only experience in work with computer. Design process by means of VLCAD is much regulated and is going on dialog mode with additional help messages. So, the main feature of VLCAD as tool for interdisciplinary researches is orientation of this system in the side of usual users, which are nospecialists in the field of information technology, instrument engineering and circuit technology. It make practicable to develop new device or verify possibility of such development by such specialists, as biologists, ecologists, medics, biochemists at el.

For such VLCAD creating, first of all, it is necessary to execute next actions:

- improve design process on the base of using mathematical methods and computer tools [Palagin et al, 1993];
- automate process of searching, processing and issuing of information;
- use methods of optimal and variant designing, effective mathematical models of design object, components and materials;
- create multi-dimensional hierarchical databases with integrated data of reference type, needed for computer-aided design;
- improve quality of designed document execution;
- increase creative part of designer work at the expense of automation of noncreative routine work;
- unify and standardize design methods;
- train specialists, including students, masters etc.;
- implement interaction with automatic systems of different levels and purposes.

To define place of VLCAD in the design process it is necessary to take into account world experience of design engineers of computer and portable devices. Integrated scheme of design process with proper outlet documentation and the place of VLCAD in design process are shown on Fig. 1. As one can see, VLCAD covers early stages of designing.

Since VLCAD has many features of CAD system it is rationally to use methodology of CAD system creating during VLCAD developing, but taking into account features of VLCAD. It is necessary to note, that now there are several conceptions of CAD system creating. Full-automatic and man-machine systems are the most widespread. First systems are difficult to build and, in some cases, it is impossible to create such full-automatic system, because design process is heterogeneous, has many internal and external connections and includes a lot of undefined factors. To take into account these undefined factors it is necessary to use creative opinion of designer.

Taking into account described above we can state, that creating of VLCAD for computer device design is very important scientific-technical problem, and implementation of such VLCAD needs certain investment. Received experience and analysis of world literature let us to separate out next main principles of such virtual laboratories creating:

1. Virtual laboratory is man-machine system. All design systems, which had been developed and now are being developed, are computer-aided, and designer is the main part of these systems. Human in such systems has to solve tasks, which cannot be well defined, and problem, which human by using own heuristic abilities may solve better and more effective than computer. Close interaction between human and computer during design process is one of principles of development and exploitation of any CAD systems for computer device designing.

2. Virtual laboratory is hierarchical system, which use comprehensive approach to automation of all design levels. Level hierarchy is presented in system structure as hierarchy of subsystems.

3. Virtual laboratory is set of informational-concerted subsystems. This very important principle refers not only to connections between large subsystems, but to connections between separate parts of subsystems. Informational compliance means, that almost all possible sequences of design tasks are served by informational-concerted programs. Two programs are informational-concerted if all data in these programs are part of numeric arrays and do not need transformations during sending from one program to another and inversely. So, results of one program can be incoming data for another program.

4. Virtual laboratory is open system, which are permanently expanding. Permanent progress of technology, designed objects, computer technology and computational mathematics lead to appearance of new, more perfect mathematical models and programs, which replace old analogs. So, VLCAD has to be open system and be able to use new methods and tools.

5. Virtual laboratory is specialized system with maximum using of unified units. Requirements of high efficiency and universality for any system are, as a rule, conflicting or competitive. It is reasonable to develop VLCAD on the base of unified parts. Necessary condition of unification is searching of common principles in the modeling, analysis and synthesis of technical objects.

Computer technology, what are offered by us, is hardware-software complex, what consist of personal computers or work stations with set of necessary peripheral items, connected in local and worldwide networks, such as Internet, and is supplied with all software. Using of these technologies already increase automation level of design stages of new devices for different purposes, including devices for interdisciplinary researches.

Today such complex systems, as VLCAD and CAD, are developed as knowledge-oriented systems, main feature of which is informational integration. Informational integration is the main application area of ontology using. Ontology, as a rule, contains hierarchy of concepts of knowledge domain and describes important features of every concept by means of mechanism "attribute–value". Connection between concepts may be described by means of additional logical statements. Constants refer to one or several concepts. This and another ontology features let to use ontology in different fields of knowledge, increasing effect from application of different methods and modes of work with information or creating on their base new more effective methods [Palagin, 2005]. Especially efficiency of ontology application can be shown in such science intensive fields, as knowledge engineering and knowledge management, objects and processes modeling, databases designing, informational integration and data mining [Gladun, 1994].

Analysis of literature and certain application domain lets to specify requirements to ontology, on the base of which VLCAD is developing [Palagin et al, 2007], [Galelyuka, 2008]:

- Ontology has to include conceptual knowledge, but not episodic ones.
- Ontology has to be specified and internal concerted with structure, names and content for all defined conceptions.
- Ontology has to be structured and simple for understanding and searching of conceptions.
- Ontology has to be limited by certain application domain for defining of used conceptions. Ontology has not to include all possible information about application domain.

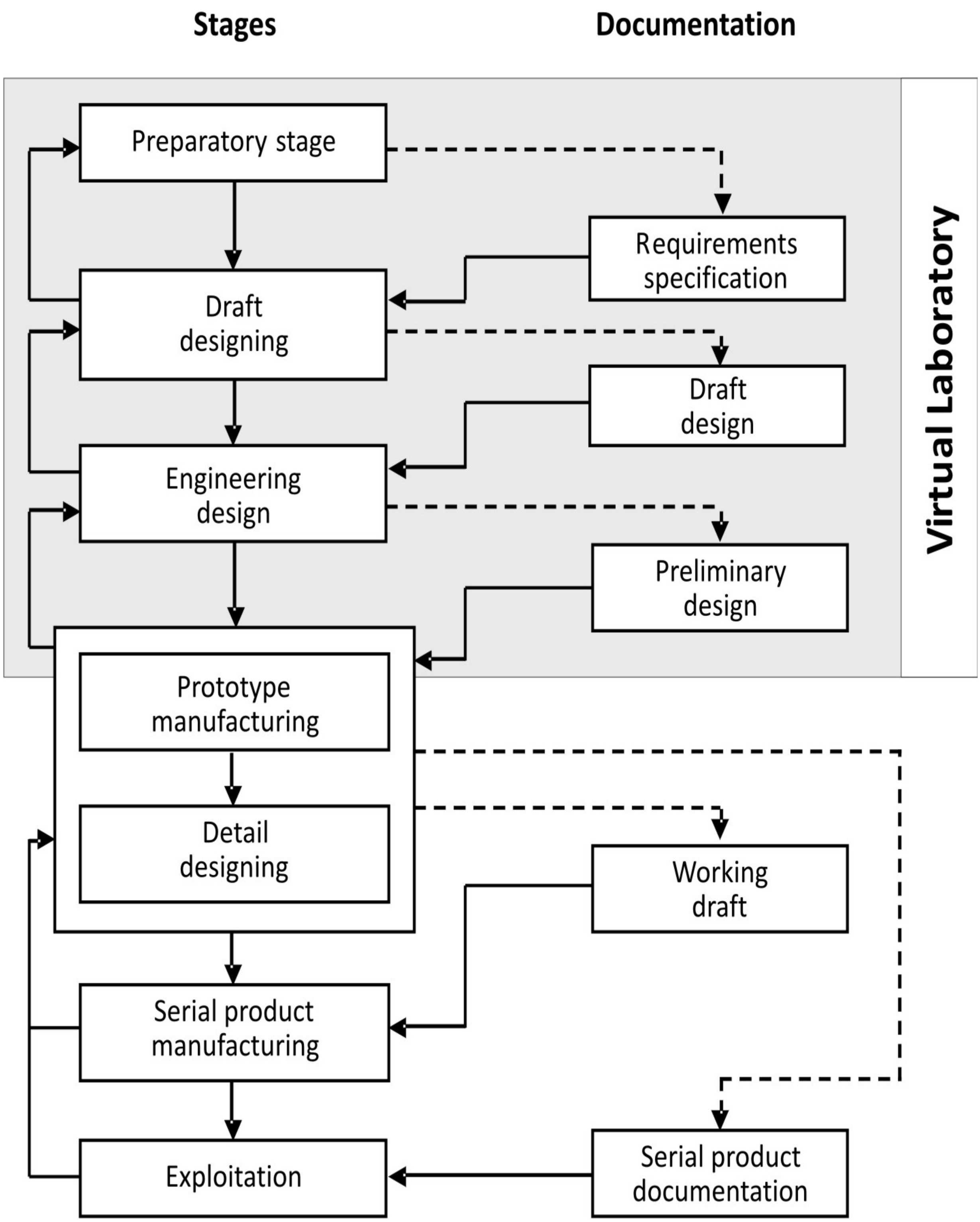


Fig. 1. Integrated scheme of design process with proper outlet documentation

VLCAD storage space

As a storage space for VLCAD a multi-dimensional access method, called ArM32, property of FOI Creative Ltd. may be used. It is built on the base of the Multi-Domain Information Model (MDIM) [Markov, 2004].

The ArM32 elements are organized in a hierarchy of numbered information spaces with variable ranges. There is no limit for the ranges the spaces. Every element may be accessed by correspond multidimensional space address given via a coordinate array.

The Multi-Domain Information Model (MDIM), presented in [Markov, 2004], is a step in the process of development of tools for data-base organization. Its main idea is to permit practically unlimited access to multi-dimensional information structures. In MDIM there exist two main constructs – numbered information spaces and basic information elements.

The Basic information element is an arbitrary long string of machine codes (bytes). When it is necessary the string may be parceled out by lines. The length of the lines may be variable. In ArM32 the length of the string may vary from zero up to 1GB. There is no limit for the number of strings in an archive but theirs total length plus internal indexes could not exceed the limit for the length of a single file of the operating system.

Basic information elements are united in numbered sets, called numbered information spaces of range 1. The numbered information space of range n is a set, which elements are numerically ordered information spaces of range n-1.

ArM32 allows using of information spaces with different ranges in the same archive (file).

The main ArM32 operations are reading, writing, appending, inserting, removing, replacing and deleting of a basic information element or any it's part.

The ArM32 numbered information spaces are ordered and main operations within spaces take in account this order. So, from given space point (element or subspace) we may search the previous or next empty or non empty point (element or subspace). In is convenient to have operation for deleting the space as well as for count its nonempty elements or subspaces.

ArM32 engine supports multithreaded concurrent access to the information base in real time.

Very important feature of ArM32 is the possibility not to occupy disk space for empty structures (elements or spaces). Really, only non empty structures need to be saved on external memory.

Using VLCAD for designing portable device "Floratest" for express-diagnostics of photosynthesis

Let remember, that only by paying attention to the design process of computer devices it will be possible to reach a high level of competitiveness of scientific developments, what lets in the future to take up notable place on the world market. It is easily to see, that most devices have the same structure, to be exact, they consist of sensor, measuring channel, data processor, interface and additional subsystems. That's why process of designing could be easily formalized. [Palagin et al, 2009]

An instance of using the VLCAD is designing the Portable Device "Floratest" as Tool for Estimating of Megalopolis Ecology State. Portable device "Floratest" is aimed for express-diagnostic of stress factors on plant state. It is developed in the V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine in the context of the program of Presidium of National Academy of Sciences of Ukraine (NASU) "Development in the field of sensor systems and technology" [Romanov et al, 2007]. The portable device measures chlorophyll fluorescence induction (CFI) without plant destruction. Using the curve of CFI (alike the cardiogram) allows diagnosing influence of one or other influential factor on the plant's state.

Party of this device is manufactured and transferred to organizations, worked in the agricultural sector, environmental protection area, mineral fertilizer production etc. for working out of methodical tools. Using of the device for estimating of megalopolis ecology state by means of evaluation of green plant state is described in the article. Together with Megalopolis Ecomonitoring and Biodiversity Research Center of National Academy of Sciences of Ukraine there were got results of experimental researches of influence detecting of heavy metals and harmful substances on the trees and plants in Kiev.

Photosynthetic processes are the processes which supply energy to the cells of plants. Chlorophyll is the main pigment of the cells of plants. One of the main features of the molecular of chlorophyll is ability of fluorescence. The intensity of chlorophyll fluorescence depends on photosynthetic activity. After irradiation of leaf the intensity of chlorophyll fluorescent signal is increasing at first and then slowly reduces. This effect is called as effect of Kautsky [Kautsky, 1931] or effect of chlorophyll fluorescent induction (CFI). The form of this curve is very sensitive to adverse environment.

It gave possibility to develop the portable device "Floratest" [Fedack, 2005, Palagin, 2007], which lets to estimate in several seconds the plant state after drought, frosts, pollution, herbicides etc. without plant damage. Like human cardiogram device builds CFI curve, which characterizes photosynthesis process, which is the base of plant vital activity.

It is possible to convert IFC curve in a description as a set of objects with features which values are integer numbers. The received description may be used by the system for inductive finding of regularities - Confor [Gladun et al, 2008], which permits to find common features of IFC curves for the trees which are under influence of equal oppresive factors. The found regularities may be used for automatic selection of harmful substances using the form of IFC curve.

In the process of designing of “Floratest” the VLCAD was used to:

- check possibility of creating of new modifications of "Floratest" on basis of developed sensors without involving specialists in circuit technology and instrument engineering at the stage of EFT-project;
- avoid expensive actual tests on the stage of device creating by replacing with virtual methods of designing and testing;
- calculate parameters (reliability, price etc.) of new device;
- prepare set of design documentations on designed device.

The sheme of using the virtual laboratory for designing of "Floratest” is shown on the fig. 2.

Principles of device operation

As a result of external influence, different objects, including biological ones, can generate plenty of radiation that is independent of these objects temperature.

All the types of radiation that were caused by some external sources of energy are called luminescence. Duration of luminescence after external influence stopping exceeds period of light fluctuations. Luminescence is conditioned by fluctuations of relatively small number of atoms or molecules of substance that become excited under energy source activity. Radiation is a result of transformation of atoms’ or molecules’ states into fundamental (unexcited) or less excited (they have less energy) states.

This is well adjusted with quantum theory, according to what every stationary orbit conforms to definite value of atom’s energy (Bore’s postulate). Being placed on stationary orbits an electron doesn’t radiate and doesn’t absorb electromagnetic waves. According to the second Bore’s postulate radiation and absorption can happen only when atom changes its state from one stationary state to another:

$$h\varpi_{mn} = h\nu_{mn} = E_n - E_m \quad (1)$$

where ϖ_{mn} or ν_{mn} – photon’s frequency, E_m, E_n – energy values of the states m and n, h – Planck's constant, m and n – the numbers of energy states. At the same time electron switches from one stationary orbit to another.

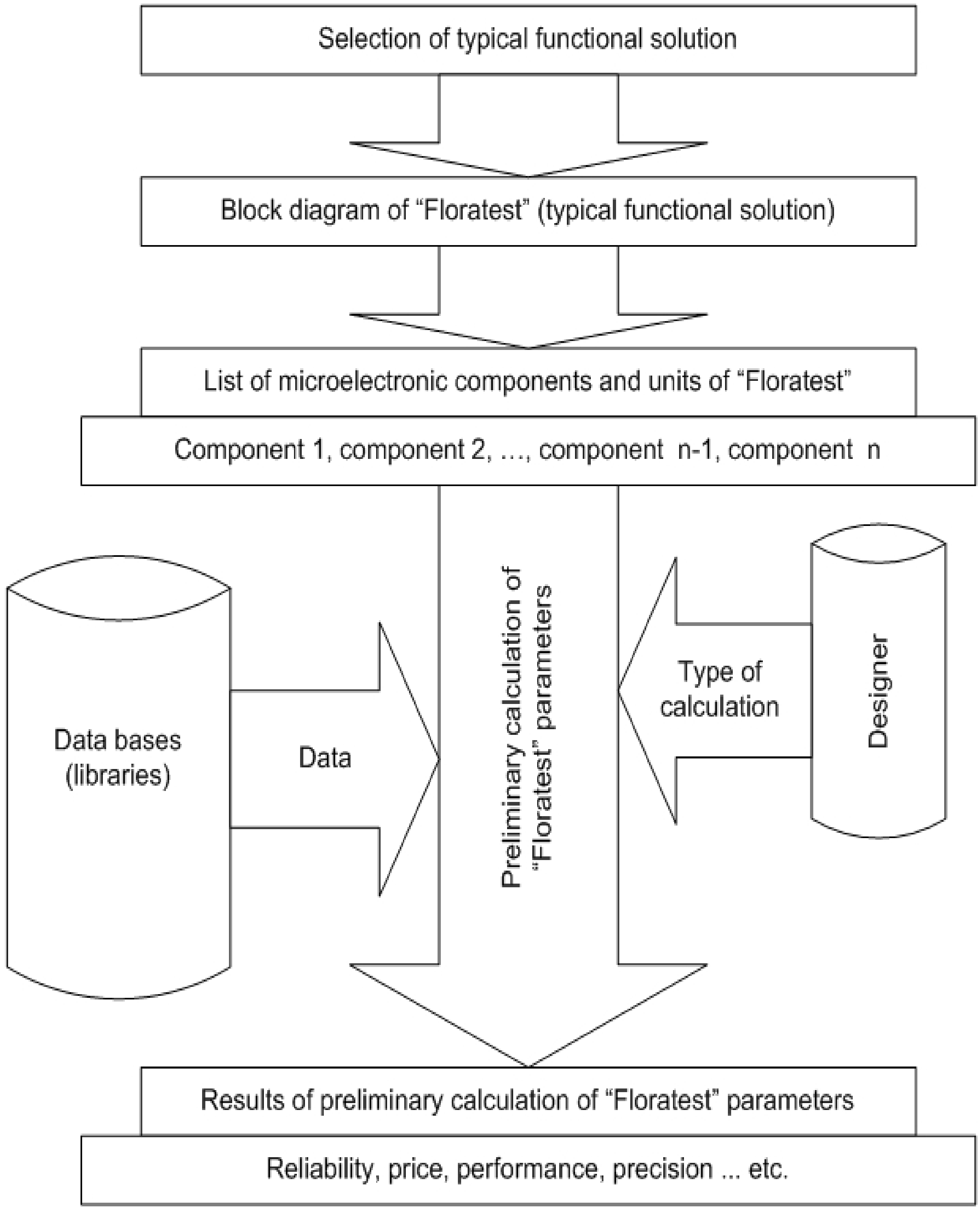


Fig. 2. Designing of "Floratest" by means of virtual laboratory

Luminescence is defined by the structure of substance energy spectrum, the average time of staying in excited states and rules of selection, which allow absorption or radiation of light of defined frequency. Short-timed luminescence is also called fluorescence. Luminescence which appears during lighting of substance (phosphor) with visible or ultraviolet light is called photoluminescence. Usually process of luminescence satisfies Stocks' rule that claims that wave length λ' of radiated light is greater than wave λ of excited light. According to the quantum theory this means that photon's energy $h\varpi(h\nu)$ is used partially for non-optical processes:

$$h\varpi = h\varpi' + E, \varpi > \varpi' \quad (2)$$

where ϖ' – luminescence's frequency, E – energy waste on another process.

Luminescence is characterized by energy output which equals to ratio of luminescence energy to energy that was absorbed by substance under stationary conditions.

Energy efficiency of photoluminescence increases proportionally to wave length λ of absorbed light up to the definite maximum value at $\lambda = \lambda_{\max}$ and then rapidly decreases to zero at $\lambda > \lambda_{\max}$ (Vavilov's rule). A sharp decrease of energy at $\lambda > \lambda_{\max}$ is explained by the fact that at these wave lengths λ the energy of absorbed photons is not enough for the process of phosphor atoms and molecules transfer to the excited states.

Ratio of luminescence photons number to absorbed photons with fixed energy is called quantum yield of photoluminescence. According to Vavilov's rule, which is under Stocks' rule, quantum yield of photoluminescence doesn't depend on wave length of excited light and rapidly decreases for anti-Stocks radiation.

Intensity of luminescence I depends on behavior of elementary processes that causes this radiation. In case of spontaneous luminescence, when radiation starts after light absorption during which atoms or molecules are transmitted to the excited level that is placed higher than the level at which radiation takes place and then these atoms (molecules) are transmitted to the luminescence level, intensity is subordinate to exponential rule

$$I = I_0 \exp(-t/\tau) \quad (3)$$

where I – lighting intensity at the moment t, I_0 – lighting intensity in a moment of excited radiation stopping, $\tau \approx 10^{-9} - 10^{-8}$ sec – an average duration of excited state of phosphor atoms or molecules. Luminescence of compound molecules and phosphorescence (after lighting) of organic substance are subordinate to the low (3).

Under influence of light there can be happened photochemical transformation of substance (including photosynthesis), which is called photochemical reactions. In a process of such reactions light absorption takes place. Energy is spent on compound molecules and polyatomic ions decomposition to component parts and creation of compound molecules of primary ones. An example of photochemical reactions is decomposition carbon dioxide under influence of light

$$2CO_2 + 2h\varpi \rightarrow 2CO + O_2 \quad (4)$$

Carbon dioxide decomposition takes place in green parts of plants under sun light influence, as photochemical process, which is a part of photosynthesis.

One of the most important properties of the molecule of chlorophyll which is the basic pigment of plant cell is ability to fluoresce. For the first time this phenomenon was researched by Kautsky [Kautsky and Hirsch, 1931]. Dependence of chlorophyll fluorescence induction on time passed after start of lightning of plant's leaves is known as an induction curve or a chlorophyll fluorescence induction curve (Fig. 3). The form of this curve is rather sensible to changes in the photosynthetic apparatus of plants during adaptation to different environmental conditions. This fact is a basic for extensive usage of Kautsky effect in photosynthesis research. The advantages of the method of CFI are the following: high self-descriptiveness, expressiveness, noninvasiveness and high sensibility.

Abilities to estimate plants states using changing of Kautsky curve form are experimentally verified. So there are examples of changing form of this curve under influence external factors.

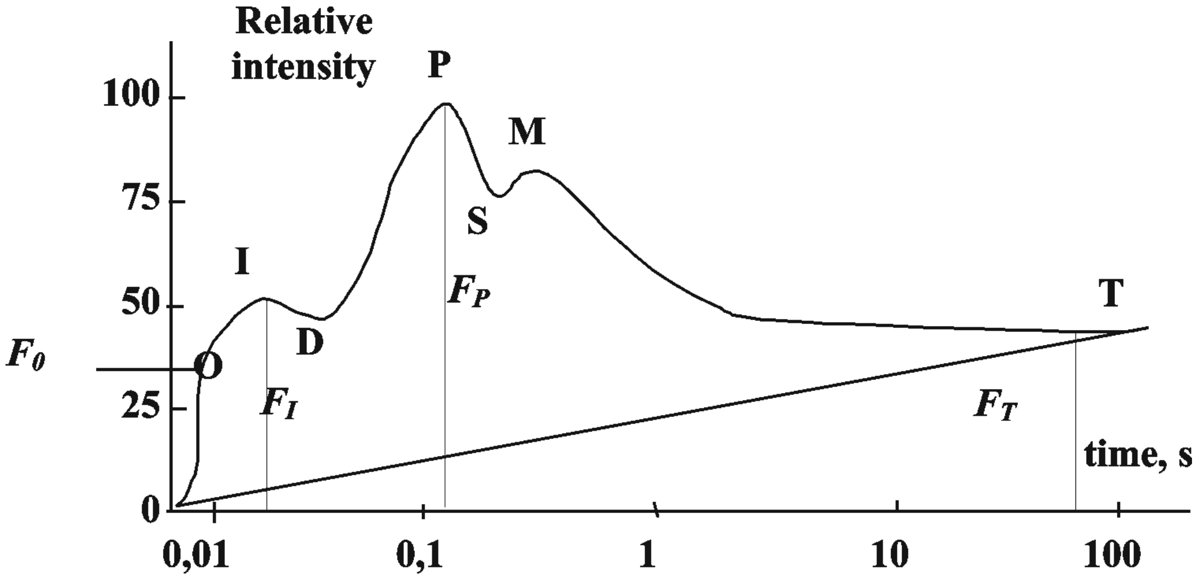


Fig. 3 Chlorophyll fluorescence induction curve

Increase of environment temperature relative to optimal for definite plant type causes decreasing of difference $F_V = F_P – F_0$. The reason is decreasing of activity of electron-transport chain or lighting activity of photosynthesis. During increasing of temperature to destruction level (45–50 °C), the level of intensity F_0 increases noticeable. It's possible to choose plants sort that are stable to high temperature influence using these parameters.

Decreasing of environments' temperature relative to optimal for definite plant type causes also decreasing of difference $F_V = F_P – F_0$, because of oppression of photochemical activity photosystem PSII. Ratio $(F_1 – F_0) / F_V$ is increasing. These features allow selecting cold-resistant plants.

Salinity of ground results in decreasing of level of F_P and F_0. Ratio $(F_P – F_0) / F_P$ is decreasing. Reason of that is oppressing of photo system PSII activity. Using these features it's possible to choose plants that are stable ground salinity.

Water deficit results in decreasing of subtraction $F_P – F_0$ in direct proportion with decreasing of water potential of leaf. Most probable reason is slowing down of photo system PSII recovery of primary acceptor because of oppression of excretion oxygen and intersystem transport of electrons.

Device features

Device and relevant diagnostic methods refer to the area of biological object researches by detecting their biophysical properties, particularly native chlorophyll fluorescent induction. Device is defined as smart biosensor with fragment plant as sensing element.

Express-diagnostic of plant state is carried out by functional features and is based on using of features of separate specific sections of IFC curve, which refer to separate areas of photosynthesis chains as diagnostic features. By IFC curve form it is easily to detect influence of one or another factor on the plant state.

Appearance of portable device "Floratest" is shown on the Fig. 4.

Fig.4. Appearance of portable device "Floratest"

Application areas of portable device for express-diagnostics of plant state:

- express-estimating of plant vital activity after drought, frosts, sorts coupling, pesticide introduction;
- express-detection of optimal doses of chemical fertilizers and biological additives, what lets to optimize amount of fertilizers and additives and reduce nitrates content in vegetables and fruits;
- express-detection of level of pollution of water, soil and air by pesticides, heavy metals and superpoison;
- economy of energetic and water resources during man-made watering;
- developing of precision agriculture technology for increasing the quality of agricultural products;
- using of the device in the insurance agriculture to get predicted results of future yield;
- automation of researches in the plant physiology field.

Functional diagram of the device is shown on the Fig.5. Data processing unit and displaying unit are built on the base of microconverter ADuC842 and graphical display with resolution capability of 128*64 pixels. Microconverter is system-on-chip for data acquisition and processing, which includes analog-digital and digital-analog converters, reference supplies, temperature sensor, timers, power supply monitor, embedded industry standard 8052 microcontroller, external and internal data memory, program memory etc.

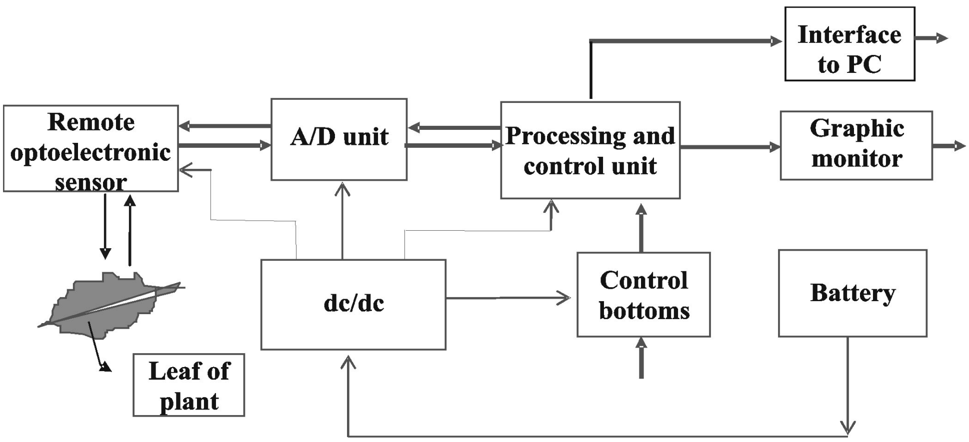


Fig.5. Functional diagram of portable device "Floratest"

Remote optical sensor is built as "reflection diagram" on the base of four light-emitting diodes and one photodetector. "Reflection diagram" means that light-emitting diodes and photodetector are situated from the same side of researched leaf. To research chlorophyll fluorescence in the red spectral region the filter is placed on the input of photodetector. Emission intensity of light-emitting diodes and photodetector sensitivity can be changed during measuring process. Integrated algorithm of device work is shown on the Fig.6.

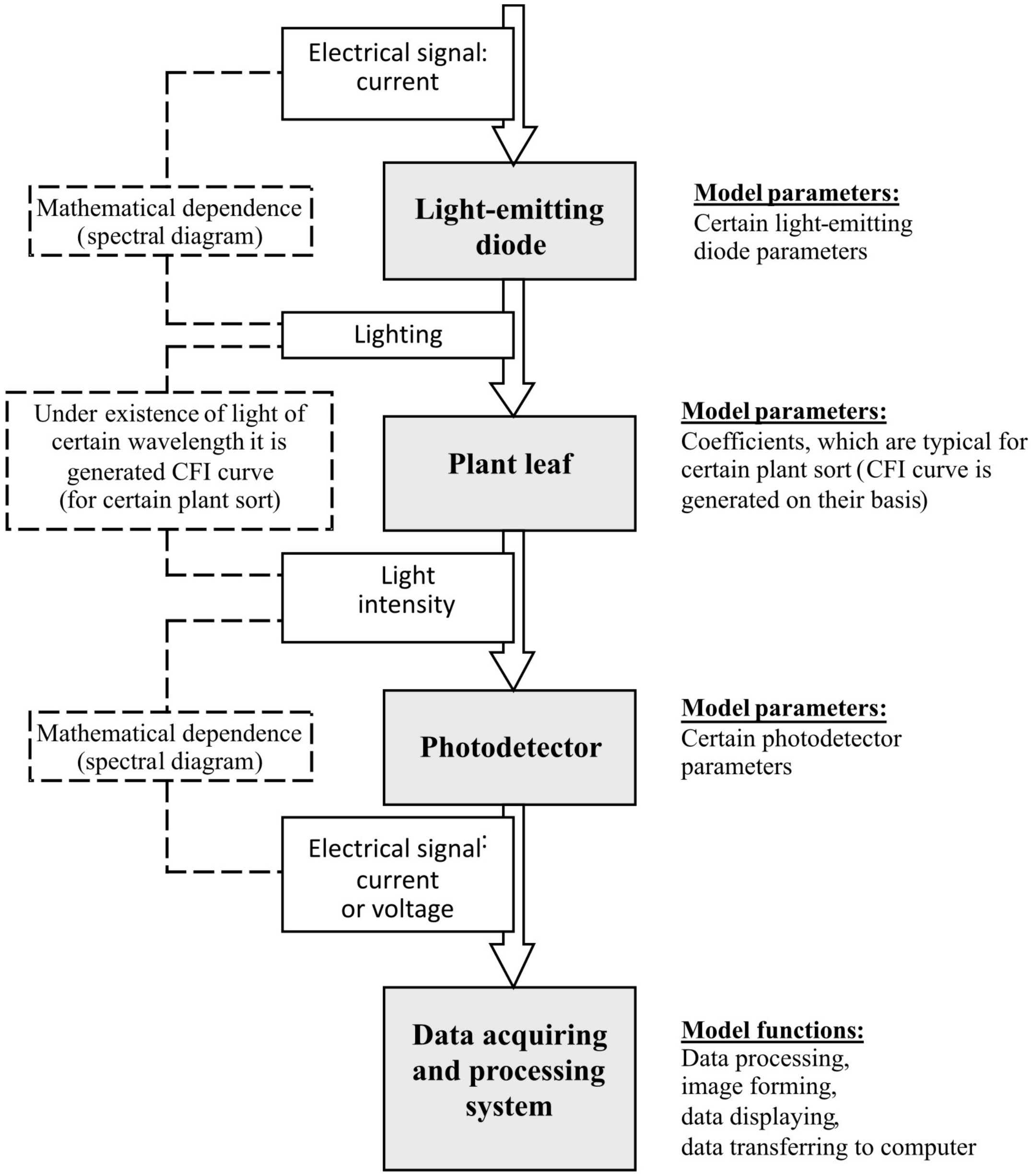


Fig.6. Integrated algorithm of device work and proper models

Today it is not enough to acquire and save measurement result in the portable device memory. It is urgent to transmit measurement results from places of measurement to laboratories or centers of operative estimation of condition and necessary decision making. For data transmitting from measuring channel to receiving point it is proposed to use mobile communication by means of midget GSM-unit with GPS-subsystem, which is embedded in the portable device, and GSM-modem, which is connected to computer or work station. During such measurements the transmissions of a small amount of data are required, so it is reasonable to use GPRS standard. Data acquiring, processing and transmitting system on the base of portable device with radio channel is shown on the Fig.7.

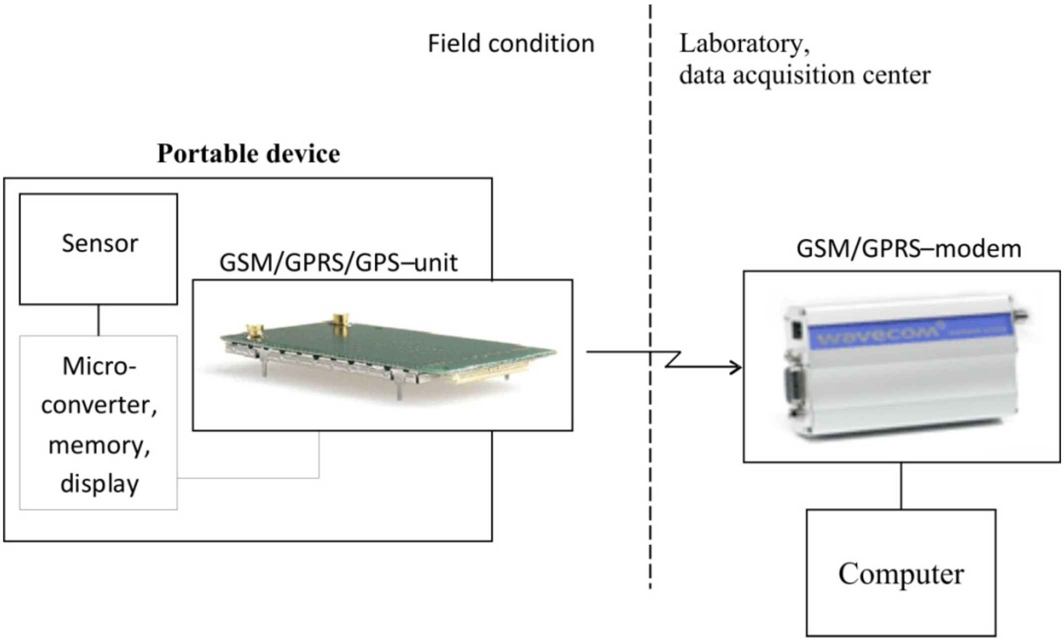


Fig.7. Data acquiring, processing and transmitting system
on the base of portable device with radio channel

Device application

The experimental researches of the "Floratest" were conducted in National Scientific Center "V.E.Tairov's Institute of viticulture and winemaking" of Academy of Agrarian Sciences of Ukraine.

Examples of the practical usage of fluorometer "Floratest" in the National Scientific Center "V.E.Tairov's Institute of viticulture and winemaking" are shown on Fig.8 and the graph of CFI on the device display are shown on Fig.9.

Fig. 8. The sensor of the "Floratest" on the vine leaf

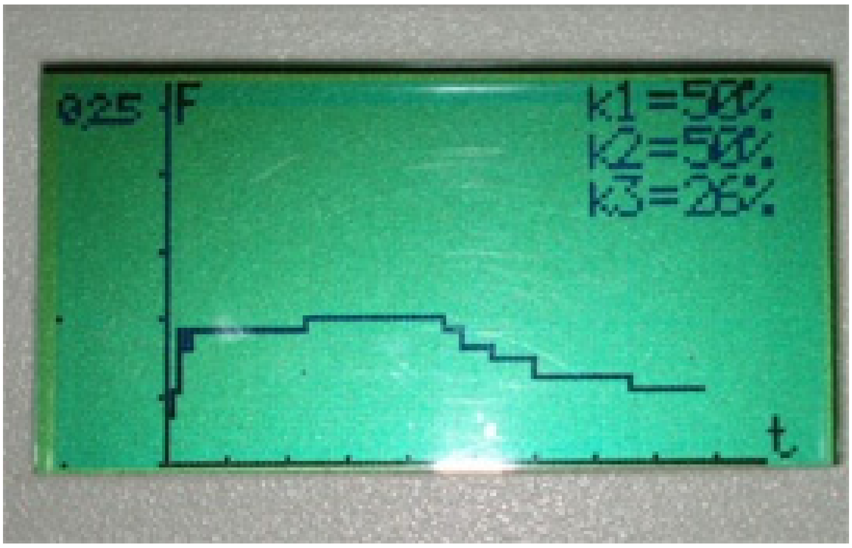


Fig. 9. The image of CFI on the device’s display

The conditions and results of the experimental researches are listed below.

Mature leaves of vine were used in the researches. Under changes of soil watering conditions there were observed sharp changes in behavior of induction transitions of chlorophyll fluorescence which were accompanied by quite essential changes of leaf tissue spectral characteristics.

Determination of fluorescence spectral characteristics was done by placing the device's sensor on the leaf's surface without integrity disturbance directly in a pot or in a field. It allowed to research on plastid and vacuolar pigments in their natural state and in that way approaching to understanding of the biophysical and physiology-biochemical processes which take place in the live leaf, and determination of important sides of photosynthetic activity.

Fluorescence intensity of the sample was determined in relative units.

It is significant that under natural conditions in the middle latitudes the drought is accompanied simultaneously by high temperatures of air, and that intensifies bad influence of ground water lack on agricultural plants.

Even in the first variant of experiment (drought) there appeared considerable changes of the behavior of fluorescence induction comparing to the control samples. Changes show in weakening of penetrability of the chloroplasts’ membrane structures. That results in substantial increase of time characteristics of fluorescence induction slow decrease. At the same time noticeable variety differences become apparent. Sharp decrease of its value is typical for profound functional injuries of photosynthetic structures and cells of particular variety entirely.

Accordingly in this stage of drought influence significant variety differences in exsiccate factor resistance of both photosynthetic structures and lamina’s parenchymal cells entirely became apparent.

More deep changes of destructive nature may be observed in case of high temperatures (+40 C), which influence on leaves complementary to drought. In this case for all the varieties being studied significant and almost irreversible functional changes of plastid structures are noted. These functional changes show in sharp decrease of CFI intensity.

Disastrous changes of life activity of vine leaf cells which take place during these processes show in oppression of biosynthetic processes, intensive decomposition of cytoplasmic structures and intensification of oxide catabolism of plant cell’s content. The consequence of these processes is decrease of CFI intensity as a result of its oxidizing transformation.

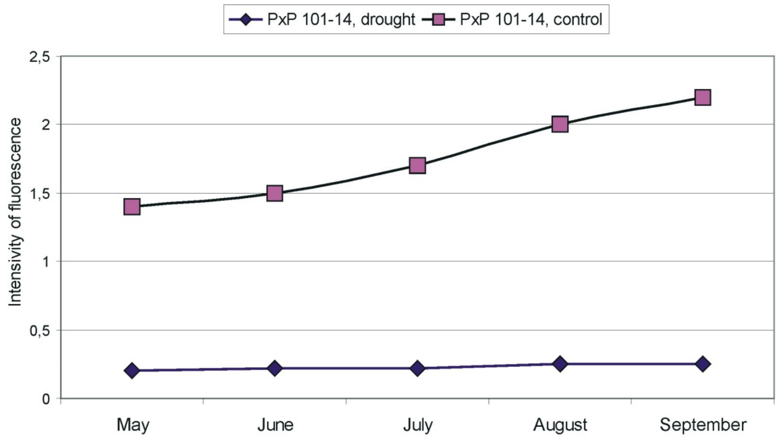


Fig. 10. CFI intensity of vine plant (sort PxP 101-14) under drought and normal conditions

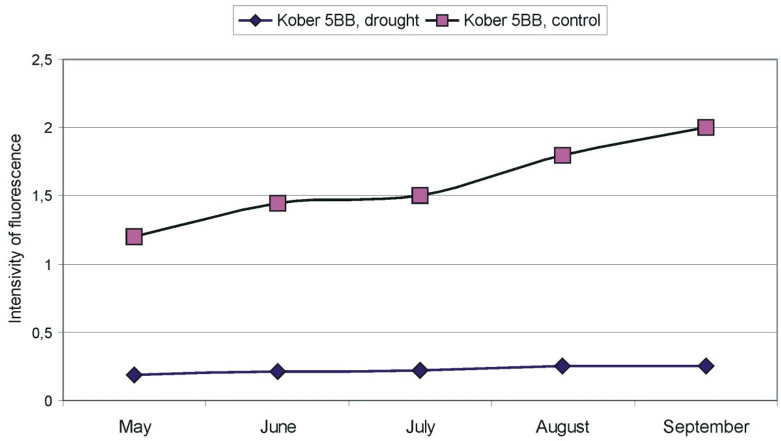


Fig.11. CFI intensity of vine plant (Kober 5BB) under drought and normal conditions

On the Fig. 10 0and Fig. 11 there are shown the diagrams of measuring of chlorophyll fluorescence intensity for two sorts of vine plants (PxP 101-14 and Kober 5BB) during 5 months. The vine plants were under drought influence and normal conditions.

Thus, a water deficit shows up on the Kautsky curve as difference of fluorescence ($F_p - F_0$) decrease. The most credible reason of this is oppression of oxygen emission which is related with slowing down of electrons transfer. Assuming that F_0 almost does not change for the test and control plants, in a maximum point the chlorophyll fluorescence intensity value can define the level of water deficit.

In 2008 together with Megalopolis Ecomonitoring and Biodiversity Research Center of National Academy of Sciences of Ukraine experimental researches of portable device "Floratest" were carried out to detect influence of heavy metals and harmful substances (e.g. lead, sodium, chlorine etc.) in leaf and soil on the plants state in Kiev. Today long-term phytomonitoring methods are used. They consist of visual observations and chemical analysis of soils and plant fragments and needs complicated equipments and lasts more than one week.

Long duration and complexity of existing methods of heavy metals and harmful substances detecting in live plants and necessity to involve skilled specialists to perform these researches set necessary conditions to develop special diagnostic methods and tools for this aim. Preliminary researches indicate that portable device "Floratest" can be used for detecting of heavy metals and harmful substances influence on state of plants by measuring of CFI curve. The form of CFI curve changes versus level of harmful substances influence.

Joint researches were carried out in Kiev green regions by means of common phytomonitoring methods and portable device "Floratest". After processing of research result there were built dependences for searching correlations between chlorine content in trees' leaf, which are got by common phytomonitoring methods (Figure 12), and readouts of portable device "Floratest" (Figure 13). Even one can see some dependence between chlorine content in trees' leaf and readouts of portable device "Floratest" (stationary region of IFC curve). Calculations, made by mathematical methods, show certain correlations between these values. Such researches were made for other harmful substances, such as sodium, magnesium.

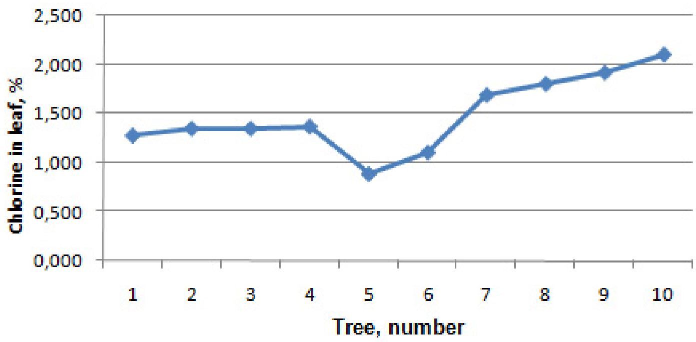


Fig. 12. Chlorine content in researched trees' leaf

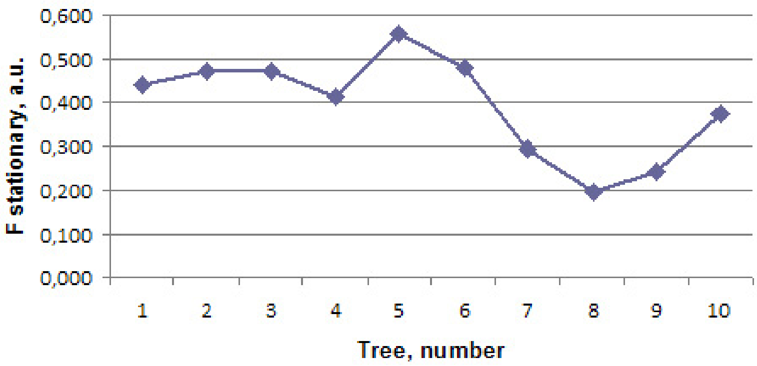
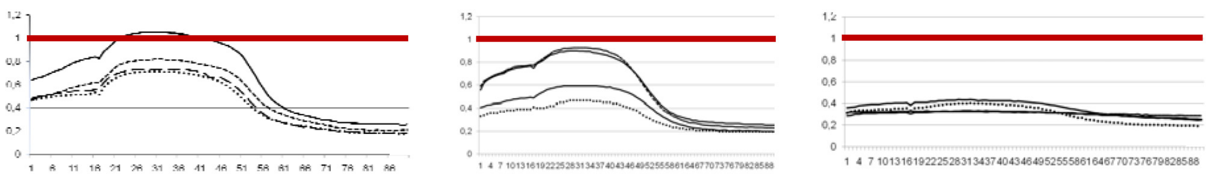


Fig. 13. Readouts for researched trees

It is easily to concede that IFC curve form expresses not only contents of separate harmful substance, but general state of tree versus influence level of harmful substances.

Researches of developing smart biosensor device "Floratest" for detecting of water deficit of plants were executed in the National scientific centre "Institute of viticulture and wine-making named after V.Ye. Tairov" of National academy of agrarian sciences of Ukraine (see Fig. 14).



(a) before predrying (b) after 10 hours of predrying (c) after 24 hours of predrying

Fig. 14. Measuring chlorophyll fluorescence induction curve

Conclusion

For increasing of competitiveness of science products it is necessary to develop new hardware-software tools, what is applicable for using in interdisciplinary researches. Virtual laboratory for computer-aided design can serves as example of such tool. In the article it is considered preconditions and main principles of such virtual laboratories creation, main purpose of which is to give possibility for sensor developers to verify ability of creating new devices on the base of their sensors on the early stages of designing, particularly on the stage of requirements specification or EFT-stage.

The features of ArM32 are appropriate for building the information base of VLCAD. The multi-dimensional information spaces make possible the effective creating of complex information structures using small amount of resources which is very important for VLCAD. At the first place the ontology's' representing and knowledge formation processes as well as intelligent recognition and classification are realizable.

An instance of using the VLCAD is designing the Portable Device "Floratest" as Tool for Estimating of Megalopolis Ecology State. Portable device "Floratest" is aimed for express-diagnostic of influence of stress factors on plant state. It is developed in the V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine.

The portable device measures chlorophyll fluorescence induction without plant destruction. Using the curve of chlorophyll fluorescence induction (alike the cardiogram) allows diagnosing influence of one or other influential factor on the plant's state. On basis of preliminary researches there were shown that using of portable device "Floratest" let to detect in express mode the worsening of photosynthetic apparatus of plant by measuring fluorescence of native chlorophyll on the early stages. During experimental researches, there were developed methodical tools, which allow evaluating the state of vine plants under drought conditions and conditions of insufficient water capacity in express-mode.

Using VLCAD the design process has been facilitated. The possibility to convert IFC curve in a description as a set of objects with features which values are integer numbers allows implementing the intellectualized components in the design process as well as in the real usage of the "Floratest".

Bibliography

[Gavrilov, 2000] Gavrylov L. Computer-aided design (CAD) systems for analog and analog-digital devices // Electronic component. – 2000. – № 3. – P. 61–66. (In Russian)

[Galelyuka, 2008] Galelyuka I. Elements of theory and tools for virtual designing of computer devices and systems of automation of biological objects experimental researches: Thesis for the candidate's degree of the technical sciences on the specialty 05.13.06 – Information technologies / I. Galelyuka. – Kiev, 2008. – 20 p. (In Ukrainian)

[Gladun, 1994] V. P. Gladun. Processes of New Knowledge Formation. Sofia, SD Pedagog 6, 1994, 192 p, (in Russian).

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

[Palagin et al, 1993] Palagin O., Denisenko E., Belycjkyy R., Sigalov V. Microprocessor system for data processing: designing and debugging / editor Beh A. – Kiev: Naukova dumka, 1993. – 352 p. (In Russian)

[Palagin and Sergiyenko, 2003] Palagin O., Sergiyenko I. Virtual scientific-innovative centers: conception of creating and perspectives of development // Control systems and computers. – 2003. – № 3. – P. 3–11. (In Russian)

[Palagin and Kurgaev, 2003] Palagin O., Kurgaev A. Problem orientation in the development computer architecture // Cybernetics and system analysis. – 2003. – № 4. – C. 167–180. (In Russian)

[Palagin, 2005] Palagin O., Yakovlev Yu. System integration of computer facilities. – Vinnitsa: Universum-Vinnitsa. – 2005. – 680 c. (in Russian)

[Palagin et al, 2007] Palagin O., Romanov V., Sachenko A., Galelyuka I., Hrusha V., Kachanovska M., Kochan R. Virtual Laboratory for Computer-Aided Design: Typical Virtual Laboratory Structure and Principles of Its Operation

// Proceeding of 4th IEEE Workshop "Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2007)". – Dortmund, Germany. – 6–8 September, 2007. – P. 77–81.

[Kautsky, 1931] Kautsky H., Hirsch A. Neue Versuche zur Kohlenstoffassimilation // Natur wissenenschaften. – 1931. – 19. – S. 964.

[Fedack, 2005] Fedack V., Kytaev O., Klochan P., Romanov V., Voytovych I. Portable Chronofluorometer for Express-Diagnostics of Photosynthesis // Proceeding of the Third IEEE Workshop on "Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications", IDAACS'2005. – Sofia, Bulgaria. – 2005, September 5–7. – P. 287–288.

[Palagin, 2007] Palagin O., Romanov V., Starodub M., Brayko Yu., Galelyuka I., Imamutdinova R., Sarahan Ye. Portable Devices for Express-Diagnostics of Photosynthesis, Viral Infections and Mycotocsicosis // Proceeding of the IV International scientific-practical conference "Urgent question and organizational-lawful aspects of Ukraine-China collaboration in the high technologies field" – Kiev, Ukraine. – 2007, 10 october. – P. 135–138.

[Gladun et al, 2008] V. Gladun, V. Velichko, Y. Ivaskiv. Selfstructurized Systems. International Journal "Information Theories & Applications". FOI ITHEA, Sofia. - Volume 15 - 2008, Number 1. -pp. 5-13.

[Palagin et al, 2009] Palagin O., Romanov V., Markov K., Velychko V., Stanchev P., Galelyuka I., Ivanova K., Mitov I. Developing of Distributed Virtual Laboratories for Smart Sensor System Design Based on Multi-dimensional Access Method. In Int. Book Series ISC book No:8: "Classification, Forecasting, Data Mining", ITHEA, Sofia, 2009, pp. 155-161.

Authors' Information

***Oleksandr Palagin** – Depute-director of V.M. Glushkov's Institute of Cybernetics of National Academy of Sciences of Ukraine, Academician of National Academy of Sciences of Ukraine, Doctor of technical sciences, professor; Prospect Akademika Glushkova 40, Kiev–187, 03680, Ukraine; e-mail: palagin_a@ukr.net*

***Volodymyr Romanov** – Head of department of V.M. Glushkov's Institute of Cybernetics of National Academy of Sciences of Ukraine, Doctor of technical sciences, professor; Prospect Akademika Glushkova, 40, Kiev–187, 03680, Ukraine; e-mail: dept230@insyg.kiev.ua, VRomanov@i.ua*

***Igor Galelyuka** – Research fellow of V.M. Glushkov's Institute of Cybernetics of National Academy of Sciences of Ukraine; Candidate of technical science; Prospect Akademika Glushkova 40, Kiev–187, 03680, Ukraine; e-mail: galib@gala.net; National University of Life and Environmental Sciences of Ukraine, Heroyiv Oborony str., 15, Kyiv, 03041, Ukraine*

***Vitalii Velychko** – Research fellow of V.M. Glushkov's Institute of Cybernetics of of NAS of Ukraine, Candidate of technical science, Prosp. Akad. Glushkov, 40, Kiev-03680, Ukraine; e-mail: glad@aduis.kiev.ua*

***Volodymyr Fedak** – Research fellow of V.M. Glushkov's Institute of Cybernetics of National Academy of Sciences of Ukraine, Prospect Akademika Glushkova 40, Kiev–187, 03680, Ukraine*

***Volodymyr Grusha** – Research fellow of V.M. Glushkov's Institute of Cybernetics of National Academy of Sciences of Ukraine, Prospect Akademika Glushkova 40, Kiev–187, 03680, Ukraine*

***Dmytro Artemenko** – software engineer of V.M. Glushkov's Institute of Cybernetics of National Academy of Sciences of Ukraine; Prospect Akademika Glushkova 40, Kiev–187, 03680, Ukraine*

***Oksana Galelyuka** – Research fellow of Institute of encyclopedic researches of National Academy of Sciences of Ukraine; Tereschenkivska str., 3, Kiev, 01004, Ukraine*

INDIRECT SPATIAL DATA EXTRACTION FROM WEB DOCUMENTS

Dimitar Blagoev, George Totkov, Milena Staneva,
Krassimira Ivanova, Krassimir Markov, Peter Stanchev

***Abstract**: An approach for indirect spatial data extraction by learning restricted finite state automata from web documents created using Bulgarian language are outlined in the paper. It uses heuristics to generalize initial finite-state automata that recognizes only the positive examples and nothing else into automata that recognizes as larger language as possible without extracting any non-positive examples from the training data set. The learning method, program realization and experiments are presented. The investigation is carried out in accordance and following the rules of EU INSPIRE Network.*

***Keywords**: Indirect Spatial Data, Automatic Data Extraction, Restricted Finite State Automata, Web Documents.*

***ACM Classification Keywords**: H.2.8 Database Applications - Data mining; F.1.1 Models of Computation - Finite State Automata*

Introduction

"Spatial data" is data with a ***direct*** or ***indirect*** reference to a specific location or geographic area [INSPIRE-DSM, 2007]. Our attention in this paper is given to the indirect references included in the text documents written in Bulgarian language.

The indirect reference to a specific location or geographic area may be of different types and formats. Because of this it is difficult to propose a common classification of such information. In the same time, one of the main characteristic of indirect references is the address, i.e. a description of the interconnection of the data with the specific location or geographic area [INSPIRE-DSAD, 2008]. Usually this is a text with common structure – location of properties based on address identifiers, usually by road name, house number, postal code, etc.

In everyday practice there are many kinds of documents containing indirect references to a specific locations or geographic areas. The kernel problem is that EU member countries use different languages and national standards for different types of indirectly given references. The automatic extraction of the references is very important for processing such documents in the INSPIRE network.

In recent years multiple machine learning approaches have been proposed for information extraction [Li et al, 2008]. A large class of entity extraction tasks can be accomplished by the use of carefully constructed regular expressions. Examples of entities amenable to such extractions include e-mail addresses, software names (web collections), credit card numbers, social security numbers (e-mail compliance), gene and protein names (bioinformatics), etc. With a few notable exceptions, there has been very little work in reducing this human effort through the use of automatic learning techniques.

In the context of information extraction, prior work has concentrated primarily on learning regular expressions over relatively small alphabet sizes and usually learning of regular expressions is done over tagged tokens produced by other text-processing steps such as POS tagging, morphological analysis, and gazetteer matching [Ciravegna, 2001].

[Rozenfield et al, 2008] propose approach, which use the immense amount of unlabeled text in the web documents in order to create large lists of entities and relations. Based on this approach the system SRES is a self-supervised web relation extraction system that learns powerful extraction patterns from unlabeled text, using short descriptions of the target relations and their attributes.

The proposed in [Li et al, 2008] learning algorithm ReLIE takes as input not just labeled examples but also an initial regular expression, which provides a natural mechanism for a domain expert to provide domain knowledge about the structure of the entity being extracted and meaningfully restriction of the space of output regular expressions.

In 2004 a team of Prof. William Cohen from Carnegie Mellon University starts creating collection of classes for storing text, annotating text, and learning to extract entities and categorize text called MinorThird [Cohen, 2004]. It contains a number of methods for learning to extract and label spans from a document, or learning to classify spans (based on their content or context within a document). The creating of such collections is a useful tool not only for the particular investigation support, but also for creating common notion for the area as a whole.

An approach for indirect spatial information extraction by learning restricted finite state automata from marked web documents created using Bulgarian language is outlined in the paper. We use heuristics to generalize initial finite-state automata that recognizes only the positive examples and nothing else into automata that recognizes as larger language as possible without extracting any non-positive examples from the training data set.

The proposed approach is a good base for building system from the class of Semi-Automated Interactive Learning (SAIL) systems [IBM, 2009]. In the next chapters the INSPIRE network as possible practical area, the proposed approach, program realization and experiments are presented. Finally, the conclusions and steps for feature work are outlined.

The INSPIRE Network

Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 for establishing an Infrastructure for Spatial Information in the European Community (INSPIRE), was published in the Official Journal of the European Union on 25 April 2007 and was entered into force on 15 May 2007 [INSPIRE Directive, 2007]. The main goal of the Directive is to establish a new common approach for processing the spatial data in all EU member countries.

A simplified view to the processing of data today is shown in the Figure 1. In most cases, each EU member state uses input data according to different, often undocumented or ill-documented data specifications and uses different methods to process the input data to produce more or less similar information relevant for policies within the Community [INSPIRE-DSM, 2007]. For instance two different states "A" and "B" have theirs own specific data specifications and data sets and their own processing methods.

The methodology described in the Directive aims for a better understanding of the common user requirements for data in INSPIRE. It focuses on the development of harmonized data specifications for the input data. This way all input data from the different member states will follow the same data specifications and the same processing steps to derive the information. The input data in the member states and their maintenance procedures will typically be more-or-less the same prior to INSPIRE, but in addition the data will be provided by the network services of the member states following the harmonized data specifications [INSPIRE-DSM, 2007]. The updated schema based on the proposed methodology for two states "A" and "B" is shown on Figure 2.

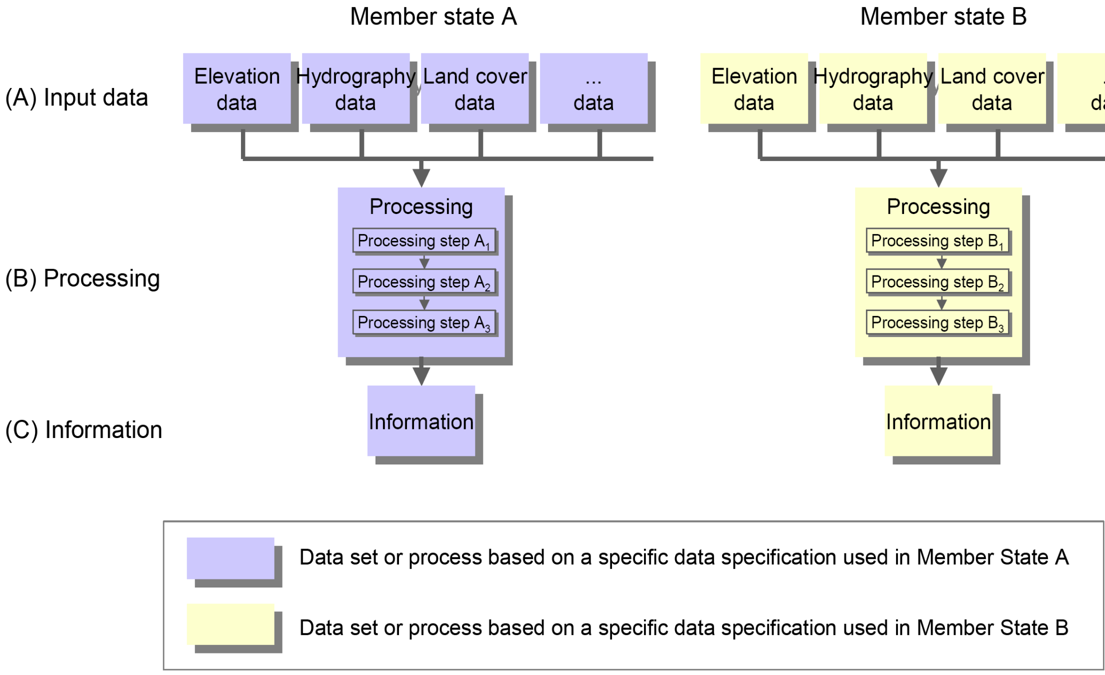


Figure 1. Current situation is "Data stovepipes"

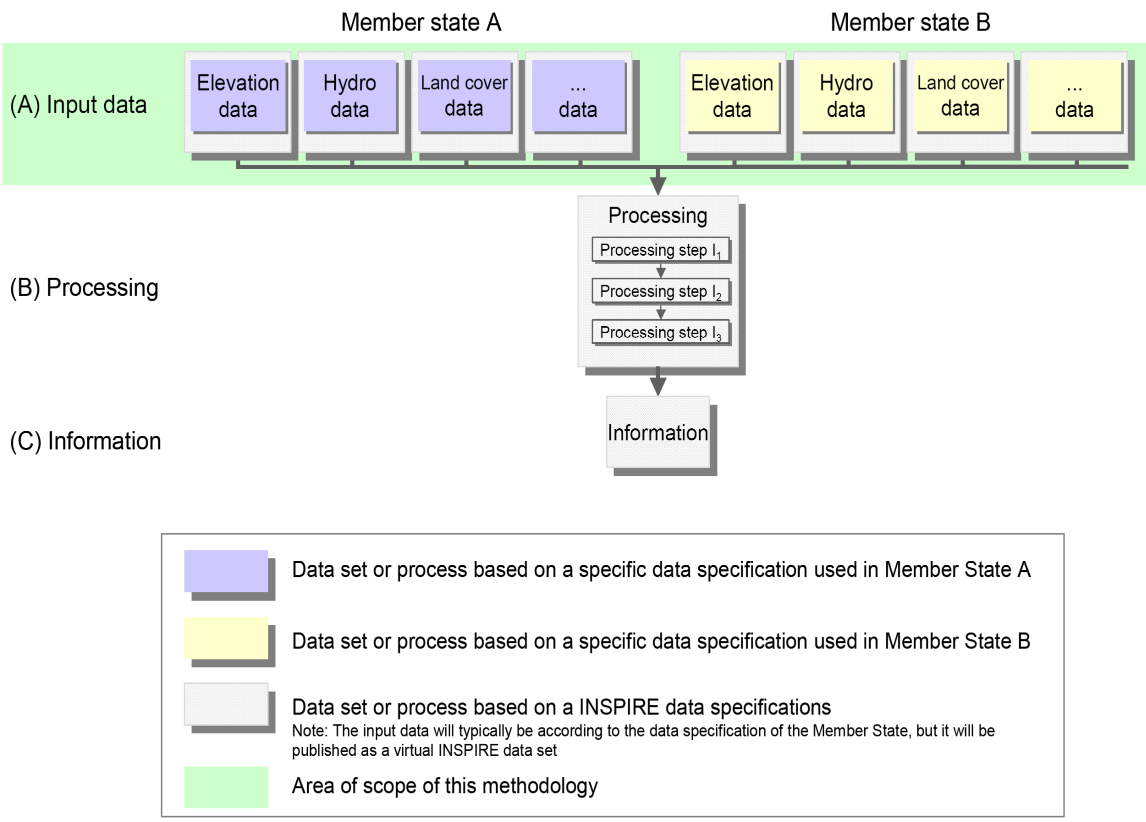


Figure 2. Target situation: Harmonized data views, eliminating data stovepipes

INSPIRE should be based on the infrastructures for spatial information that are created by the member states and are designed to ensure that:

- spatial data are stored, made available and maintained at the most appropriate level;
- it is possible to combine spatial data from different sources across the European Community in a consistent way and share them between several users and applications;
- it is possible for spatial data collected at one level of public authority to be shared between other public authorities;
- spatial data are made available under conditions which do not unduly restrict their extensive use;
- it is easy to discover available spatial data, to evaluate their suitability for the purpose and to know the conditions applicable to their use.

For these reasons, the Directive focuses in particular on five key areas:

- metadata;
- the interoperability and harmonization of spatial data and services for selected themes (as described in Annexes I, II, III of the [INSPIRE Directive,2007]);
- network services and technologies;
- measures on sharing spatial data and services;
- coordination and monitoring measures.

INSPIRE lays down the legal framework for the establishment and operation of an Infrastructure for Spatial Information in Europe. The purpose of such an infrastructure is to assist policy-making in relation to policies that may have a direct or indirect impact on the environment. *"Infrastructure for spatial information"* means metadata, spatial data sets and spatial data services; network services and technologies; agreements on sharing, access and use; and coordination and monitoring mechanisms, processes and procedures, established, operated or made available in accordance with the Directive [INSPIRE Directive, 2007]:

Every spatial object in a spatial data set needs to be described by a data specification specifying the semantics and the characteristics of the types of spatial objects in the data set. The spatial object types provide a classification of the spatial objects and determine among other information the properties that any spatial object may have (be they thematic, spatial, temporal, a coverage function, etc.) as well as known constraints (e.g. the coordinate reference systems that may be used in spatial data sets). This information is captured in an application schema using a conceptual schema language, which is a part of the data specification. As a result, a data specification provides the necessary information to enable and facilitate the interpretation of spatial data by an application [INSPIRE-TAO, 2007].

The logical schema of the spatial data set may and will often differ from the specification of the spatial object types in the data specification. In this case, and in the context of real-time transformation, a service will transform queries and data between the logical schema of the spatial data set and the published INSPIRE application schema on-the-fly. This transformation can be performed by the download service offering access to the data set or a separate transformation service.

*The main goal of INSPIRE is the "**Interoperability**"* which means the possibility for spatial data sets to be combined, and for services to interact, without repetitive manual intervention, in such a way that the result is coherent and the added value of the data sets and services is enhanced.

One important aspect of this process is the automatic extraction of spatial data and creating the corresponded metadata.

Data Extraction by Learning Restricted Finite State Automata

The approach for indirect spatial information extraction by learning restricted finite state automata from marked web documents contains four main steps:

1. Setting up the hierarchical structure of the data to be extracted. Every element and sub-element which is to be identified has to be specified. The data structure is expressed as a tree of elements and their sub-elements.
2. Scanning and manual tagging the initial documents for the required information.
3. Extracting the examples for the different elements and building an initial parsing grammar.
4. Data extracting from new documents. The user can continue to improve the accuracy of the results by manually correcting the annotations for a particular document and add it to the learning set.

The building of the parsing grammar consists of two sub-steps:

a) combining all positive examples;

b) generalizing the resulting tree into restricted finite state automata.

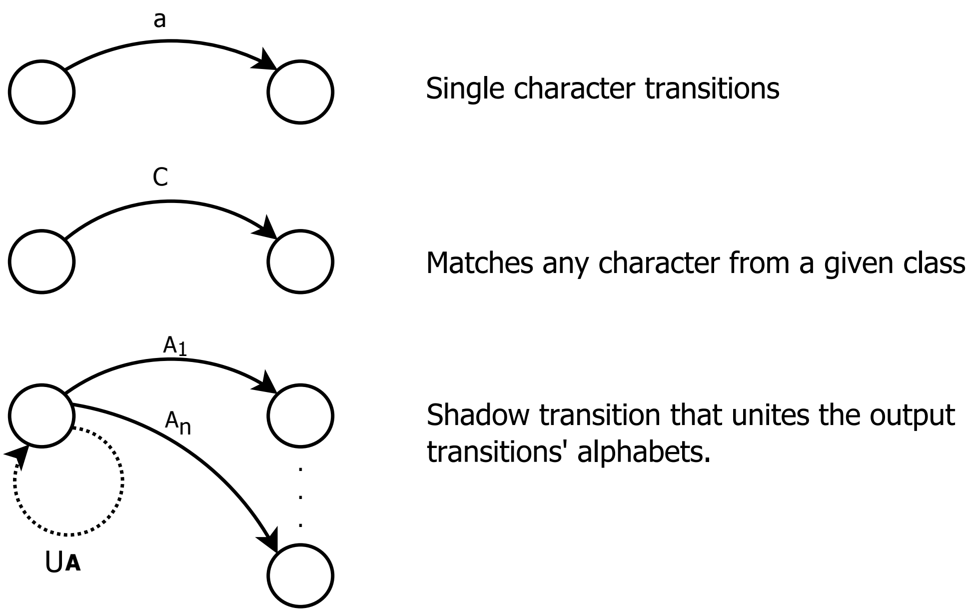


Figure 3. Elements of the restricted finite state automata [Baltes, 1992]

At the sub-step a) all marked instances of the structured data are flattened in strings containing the text of the main element with special symbols marking the beginnings and ends of the sub-elements and the HTML tags in the case when the text is a web document. Then all the flattened strings from all documents are combined in a single tree. This tree is then used as the initial finite state automata. It recognizes all learned positive examples without misrecognizing negative ones.

The sub-step b) is the generalization of the automata using heuristic methods for combining states and extrapolating the transitions' characters into predefined alphabets. After each generalization the automata is checked for consistency by re-scanning the learning texts and if the extracted data differs from the initial (manually annotated) data the modification is rolled back. There are many ways in which the finite state automata can be generalized [Baltes, 1992]. To prevent the computational complications that arise from this condition we use restricted finite state automata. The building elements that are used in these automata are (Figure 3):

- single character transition;
- matching any character from a given class;
- shadow transition that unites the output transitions' alphabets.

In addition to the automata generalization heuristics described later in the section the generalizator employs the use of a custom character class list. The class list specifies what characters belong to a given class and how many of them have to be present in a state's output transitions before class generalization is attempted. Table 1 shows one sample list which includes classes for both English and Bulgarian letters.

Table 1: Sample character class list

Min	Characters	Class
3	abcdefghijklmnopqrstuvwxyz	English lowercase
3	ABCDEFGHIJKLMNOPQRSTUVWXYZ	English capitals
3	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ	English
3	абвгдежзийклмнопрстуфхцчшщъьюя	Bulgarian lowercase
3	АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЬЮЯ	Bulgarian capitals
3	абвгдежзийклмнопрстуфхцчшщъьюяАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЬЮЯ	Bulgarian
1	0123456789	Digits
1	\b \t \n \r	White space characters
1	' " " „ " « » ' ' ‚	Quotes

The generalization algorithm (Figure 4) in sub-step b) is done in the following way:

1. Class generalization (top-down) which tries to generalize as much as possible output transition characters for a given state into classes;
2. State merging (bottom-up) with character comparison tries to merge a state with one of its next possible states if the two states have identical characters and classes on their output transitions. If it is successful, the two states are merged into a single state and a shadow transition is added over the union of the other output transitions' alphabets. Further testing is made to find the upper repetition limit for the newly formed state;
3. State merging (bottom-up) without character comparison, essentially same as above except it does not require two states to have comparable characters and classes on their output transitions. If it is necessary, character transitions are merged with class transitions. This operation is more prone to making erroneous generalizations or one that block the further generalization of upper states therefore it is performed after the previous generalizations;

4. Character and class merging (bottom-up) tries to merge a character transition in a given state with a class or another character transition in the same state resulting in a transition over a new class which was not predefined in the classes list;
5. State skipping (bottom-up) which tests if all output transitions on a given state can be skipped thereby advancing onto all sub-states without matching any of the transitions. Every output transition to another state is complemented with an epsilon transition (one that matches the zero-length string) to the same sub-state.

After every change of the automata a test run is performed with the new automata over the learning data set. If the result differs from the initial ones the attempted transformation is rolled back. All generalization and merging steps (without state skipping) are repeated until there are no more states which can be merged and/or generalized.

Once the module's learning phase is complete a parsing grammar is being generated. This grammar employs regular expressions for data extraction and generates structured XML output containing all elements found in the parsed documents. The sub-elements of the hierarchical data structure are encoded as named groups in the regular expressions. This grammar can then be used for performing background batch processing on a large number of documents or to analyze the produced regular expressions and make inferences for the structure of the elements of interest.

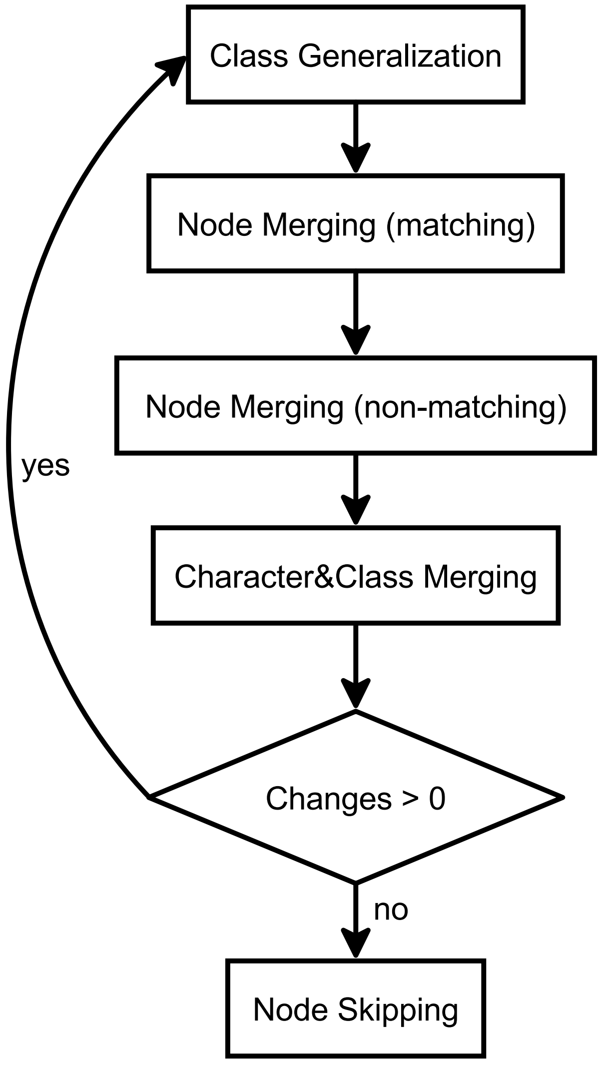


Figure 4. Generalization algorithm

Program Realization

The presented algorithm has been realized in the experimental system InDES. The system contains two separate modules: a graphical user interface (GUI) and a command-line learning and extracting module. The GUI is developed in C#.NET and employs the embedded Internet Explorer browser component to display the web documents. The learner and extractor are written in C++ for increased performance and smaller memory footprint. Both modules use the same html preprocessing routine for cleansing the given web documents. The cleansing's purpose is to normalize or eliminate characters in the input document without changing the structure of the contained information or the way it appears on the screen. This includes but is not limited to Unicode character normalization where explicit character codes are replaced with their respective characters and JavaScript removal (since the current version of the system does not execute JavaScript prior to learning or extracting).

A screenshot of the GUI with a loaded web document is given on Figure 5. In the left, the sub-screen for selecting the web documents contain some already connected web sites and corresponded documents. At the right the generated grammar and founded matches are shown. In the center of the screen the current document with market texts is presented.

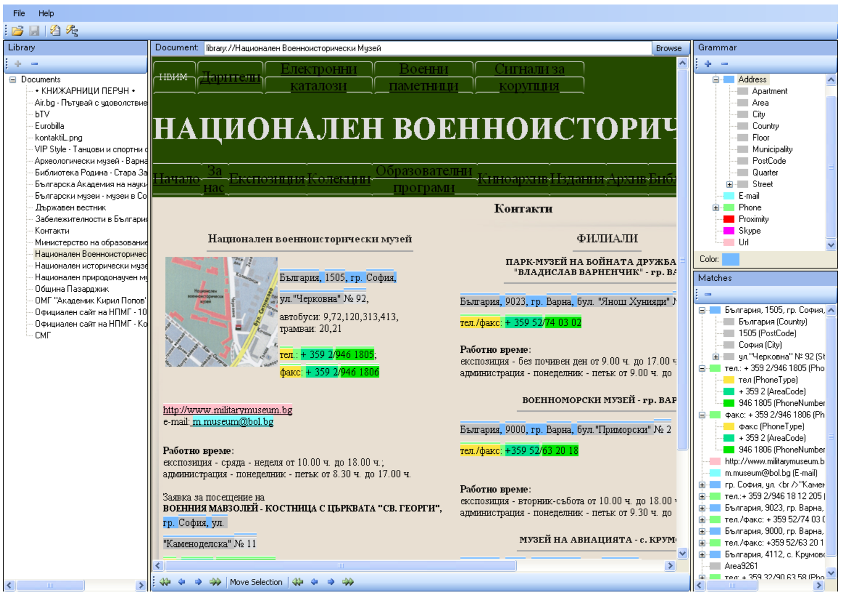


Figure 5. Screenshot of the program system InDES

Experiments

We have made several types of experiments over different web documents in Bulgarian language for extracting elements such as addresses, phones and e-mails from randomly chosen companies' and social institutions' web pages containing contact information. To test the ability of the proposed method to extract new information we used a set of 100 pre-marked web documents in Bulgarian language for three types of elements: addresses, phones and e-mails with corresponded sub-elements. The reason for such choice is that main way for representing the indirect space information is using the addresses [INSPIRE-DSAD, 2008]. In other words, the extracting of the address is the first step for the processing the indirect space information. This information may contain different elements, represented by text sequences which are connected to any specific location or geographic area by the addresses. These elements need to be extracted, too. This means that the system need to have possibility to extract addresses as well the other types of elements from the given thematic area. To simplify the experiments, the phone numbers and e-mails are taken as such elements.

The experiments were provided following the steps of the proposed approach.

At first step the hierarchical structure of the data to be extracted was set up as it is shown on Figure 6. The structure consists of:

- addresses with sub-elements "Country", "Area", "Municipality", "City", "Post Code", "Quarter", "Street", "Floor" and "Apartment";
- phones with sub-elements "Area Code", "Phone Number" and "Phone Type";
- e-mails without sub-elements.

At the next step the data set was chosen. For the purposes of the experiments, the web document set was created using web pages for five categories organizations: companies, schools, museums, municipalities and libraries. The documents were picked out in html format using Google possibilities. For each category were selected first twenty web sites after searching for combination keywords address and one of keywords "company", "school", "museum", "municipality", "library" and with restriction "pages in Bulgarian".

Then, all documents from the data set was scanned and manually tagged in accordance with chosen hierarchical structure. Some of the documents are used later as instances in the learning set and other are used as instances in the testing set. At the first experiment we used ten-fold cross validation. At the second experiment the data set was split into learning set and testing set in random principle.

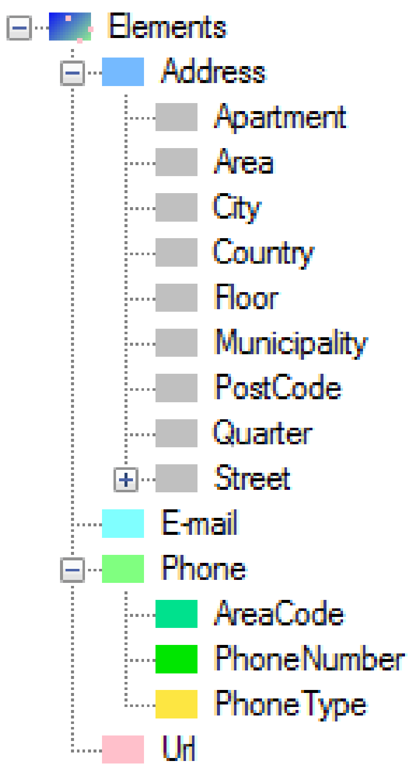


Figure 6. Sample element hierarchy
for information extraction

Since the task is to find and extract all data that represents a given element we tested the system using the following criteria:

- Recall – the percentage of manually annotated elements for which an overlapping element is found in the results of the search;
- Precision – the percentage of found elements that overlap with a manually annotated element [Taylor, 1982];
- Accuracy – the average similarity between the original annotated elements and the correctly extracted elements.

For a similarity measure we propose to set the ratio of the length of the overlapping to the length of the union of the original and extracted texts:

$$similarity = \frac{A \cap B}{A \cup B}$$

where A and B are the original and extracted line segments respectively.

1. Ten-fold cross-validation test

Because of the wide diversity in which those elements can occur we split the data into 10 parts and performed a 90% learn – 10% test evaluation testing once each part [Kohavi, 1995]. Table 2 shows the results for each of the three element types.

Table 2: Results for extracting addresses, phones and e-mails without sub-elements

	Count	Recall	Precision	Accuracy
Address	134	51.54%	74.56%	57.25%
Phone	296	82.71%	87.45%	69.41%
E-mail	102	89.96%	97.29%	95.44%

During generalization the number of states in the automata has been reduced on average by 71%, 72% and 90% for addresses, phone numbers and e-mails respectively. The automata could be further compacted by merging common sub-trees.

This experiment shows the ability of the learning method to build generalized automata for parsing web documents. There appears to be a relation between the algorithm's performance and the structural variance of the information to be extracted.

2. Examination trend of reaching satisfactory results with increasing the cardinality of learning set

In other group of experiments the data set was split in two parts in a random principle – 40 instances were used as a learning set and the rest 60 documents were used as a testing set.

The system was learned using respectively 10, 20, 30 and 40 web documents from the learning set (each set contained the documents of the lower learning sub-set). Each time the testing was provided with all documents from the testing set. The test results were analyzed to obtain values for the numbers of fully extracted, partially extracted and elements that should have been but were not extracted.

These experiments were provided in order to examine the trend of reaching satisfactory results. For each case multiple randomized runs have been performed to obtain more stable average values. We assume the address is recognized if its sub-elements, given in the text, are recognized. The telephone number is recognized if the system has recognized at least the phone code and the number. In several cases the system has recognized the string as a whole without recognizing its sub-elements. Partial recognizing means that some sub-elements are recognized (in particular one of them), but not the element as a whole. For instance only "town", "street", etc.

Table 3 shows the obtained results.

Table 3: Precision for extracting addresses, phones and e-mails
when learning sets were 10, 20, 30 and 40 documents respectively

Number of Learning Documents	Address	Phone	E-mail
10	45.33%	85.54%	93.30%
20	50.43%	81.17%	86.72%
30	54.24%	84.23%	88.73%
40	66.19%	85.57%	93.35%

Figures 7, 8 and 9 shows the trend of increasing learning accuracy with increasing of the cardinality of learning set for elements and sub-elements of addresses and phones respectively.

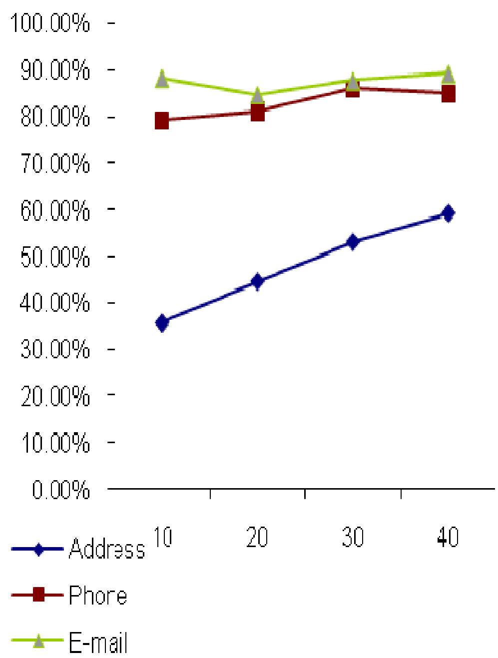


Figure 10. F-measure for addresses, phones and e-mails

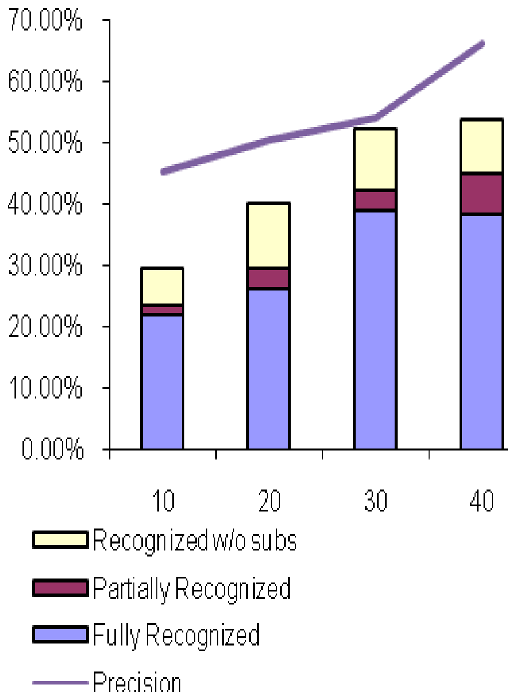


Figure 11. Recall and precision for the addresses

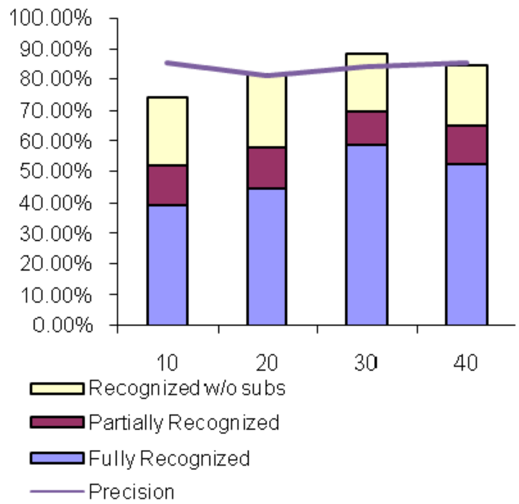


Figure 12. Recall and precision for the phones

In this experiment we found there is a trend for increasing the accuracy of the extractor by increasing the learning data set. As expected, e-mail addresses show the highest recall and precision and achieve high accuracy with a small cardinality of the learning set. The main reason for it is probably the existence of a strict and short structure for an e-mail address which leads to little variety in the different element instances. In that case learning over wider range of documents can actually sometimes prevent the optimal generalization resulting in worse results (Table 3). Bulgarian addresses show the worse results. Given the extremely wide variety of the indirect representation of Bulgarian addresses the results for this element are very promising. Furthermore, by increasing the learning and testing data sets the automata should begin to comprise of the most common cases which will lead to results comparable to the ones for the other two elements.

Our results are compatible with [Cohen, 2004]. The differences are coming from different languages and different grammars structure in the languages.

Conclusion

The aim of the current work was to propose an approach for indirect spatial data extraction by learning restricted finite state automata from marked web documents. The learning method, program realization and experiments were presented. The proposed approach is suitable to cover practical needs for automatic extraction of indirect spatial data from web documents created using Bulgarian language.

The developed system InDES and provided experiments showed that such approach is acceptable and can be used in INSPIRE network.

The main idea of the approach is based on the understanding that the most of indirect spatial information objects are referenced to specific locations or geographic areas using the addresses. In near future, such kind of information will be given following the INSPIRE Data Specification of Addresses [INSPIRE-DSAD, 2008]. It is good standard which is accepted all over the European Community and need to be basis for the further investigation.

The future work involves research in the following directions:

- Adding external knowledge to the system (part-of-speech tagging, named entity lists, word ontology);
- Enhancing the generalization algorithm to identify common sub-trees and merge them if possible;
- more detailed comparison the performance of the realized system with other existing systems like MinorThird which implements various other extractor learning algorithms [Cohen, 2004].

Acknowledgements

This work is partially supported by Bulgarian National Science Fund under the project **D 002-308 / 19.12.2008** "Automated Metadata Generating for e-Documents Specifications and Standards".

Bibliography

[Baltes, 1992] J. Baltes. Symmetric Version Space Algorithm for Learning Disjunctive String Concepts. Technical Report 92/469/06, University of Calgary, Calgary, Alta, March 1992

[Ciravegna, 2001] F. Ciravegna. Adaptive Information Extraction from Text by Rule Induction and Generalisation. IJCAI 2001, pp. 1251-1256.

[Cohen, 2004] W. Cohen. Minorthird: Methods for Identifying Names and Ontological Relations in Text Using Heuristics for Inducing Regularities from Data. 2004. http://minorthird.sourceforge.net

[Holzmann, 1991] J. Holzmann. Design and Validation of Computer Protocols. Prentice Hall, 1991, 512 p.

[IBM, 2009] Trainable Information Extraction Systems. http://researchweb.watson.ibm.com/IE/

[INSPIRE Directive, 2007] Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE) http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF

[INSPIRE-DSAD, 2008] INSPIRE Data Specification of Addresses http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/DataSpecifications/INSPIRE_DataSpecification_AD_v2.0.pdf

[INSPIRE-DSM, 2007] INSPIRE Data Specifications: Methodology for the Development of Data Specifications http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/inspireDataspecD2_6v2.0.pdf

[INSPIRE-TAO, 2007] INSPIRE Technical Architecture Overview. http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/INSPIRETechnicalArchitectureOverview_v1.2.pdf

[Kohavi, 1995] R. Kohavi. A Study of Cross Validation and Bootstrap for Accuracy Estimation and Model Selection. International Joint Conference on Artificial Intelligence IJCAI, 1995.

[Li et al, 2008] Y. Li, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, H. Jagadish. Regular Expression Learning for Information Extraction. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, October 2008. Association for Computational Linguistics, pp. 21–30.

[Taylor, 1982] J. Taylor. An Introduction to Error Analysis. University Science Books, Mill Valley, California, 1982.

[Rozenfield et al, 2008] B. Rozenfield, R. Feldman. Self-supervised relation extraction from the Web. Knowledge and Information Systems, Volume 17, Issue 1, (October 2008), Springer-Verlag New York, Inc. USA, pp. 17-33.

Authors' Information

Dimitar Blagoev *– Plovdiv University "Paisii Hiledarsk"i, PhD Student in Computer Informatics Department; 24, Tsar Asen St., 4000 Plovdiv, Bulgaria; e-mail: gefrix@gmail.com*

George Totkov *– Plovdiv University "Paisii Hiledarski"; chair of Computer Informatics Department; 24, Tsar Asen St., 4000 Plovdiv, Bulgaria; e-mail: totkov@uni-plovdiv.bg*

Milena Staneva *– Institute of Mathematics and Informatics – BAS; Information Systems Department; Acad. G.Bontchev St., bl.8, Sofia-1113, Bulgaria; e-mail: mstaneva@math.bas.bg*

Krassimira Ivanova *– Institute of Mathematics and Informatics – BAS, Information Systems Department; Acad. G.Bontchev St., bl.8, Sofia-1113, Bulgaria; e-mail: kivanova@math.bas.bg*

Krassimir Markov *– Institute of Mathematics and Informatics – BAS, Information Systems Department; Acad. G.Bontchev St., bl.8, Sofia-1113, Bulgaria; e-mail: markov@foibg.com*

Peter Stanchev *– Kettering University, Flint, MI, 48504, USA / Institute of Mathematics and Informatics – BAS; chair of Information Systems Department; Acad. G.Bontchev St., bl.8, Sofia-1113, Bulgaria; e-mail: pstanche@kettering.edu*

TABLE OF CONTENTS OF IJ ITK, VOLUME 3

Number 1

Number 2

Number 3

Number 4

INDEX OF AUTHORS OF IJ ITK, VOLUME 3

TABLE OF CONTENTS OF VOLUME 3, NUMBER 4