ITHEA

International Journal INFORMATION TECHNOLOGIES & KNOWLEDGE

2009 Volume 3 Number 4

International Journal
INFORMATION TECHNOLOGIES & KNOWLEDGE

Volume 3 / 2009, Number 4

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)**

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.

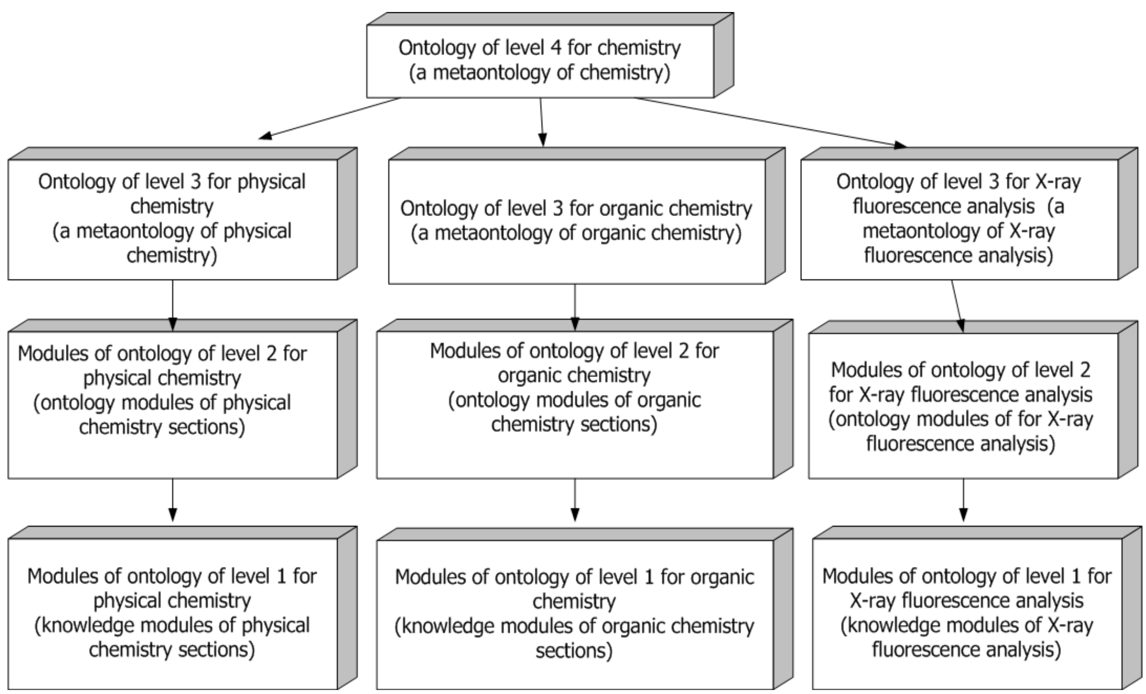

Fig. 1. The structure of multilevel ontology of chemistry

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.

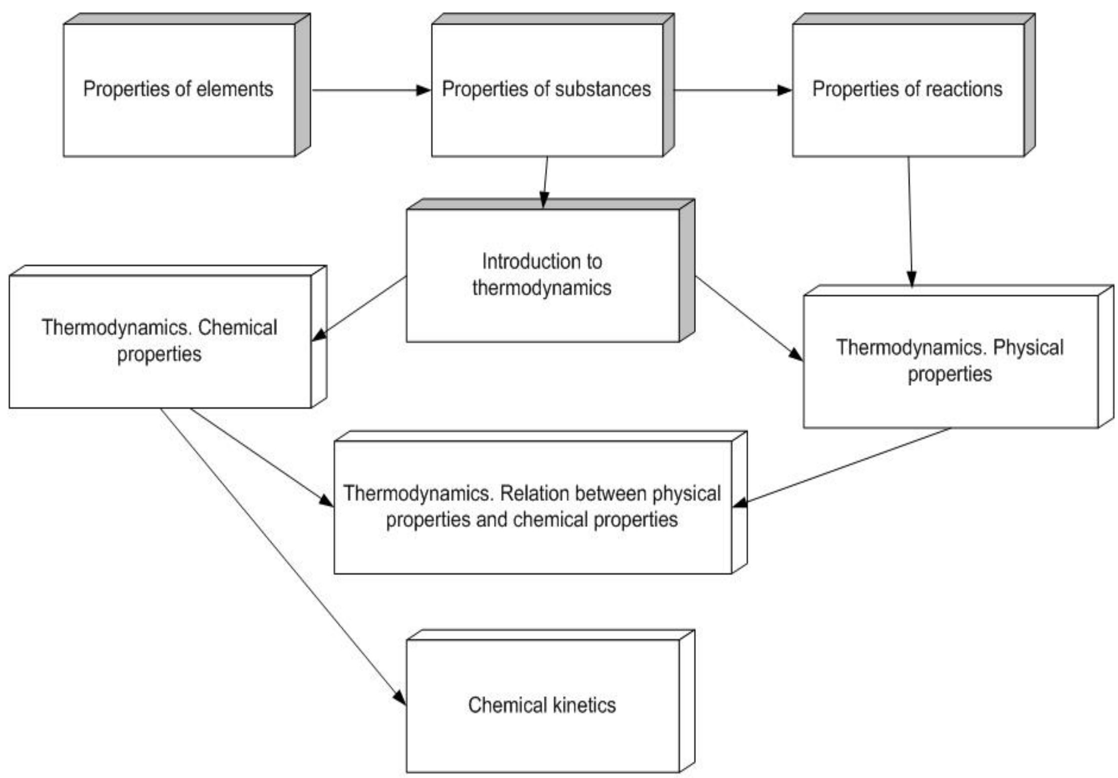

Fig. 2. The modules of the ontology of level 2 for physical chemistry

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.

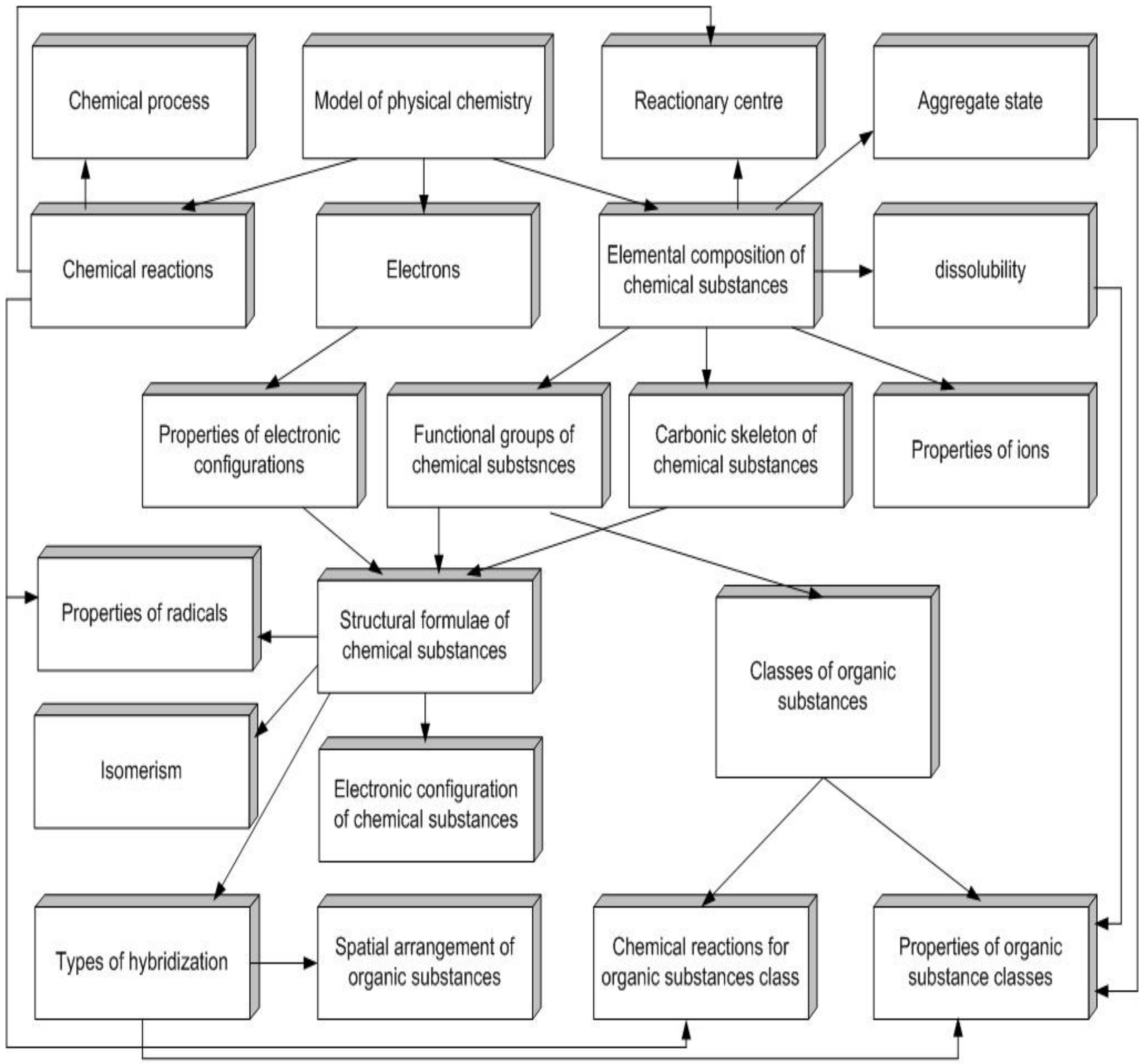

Fig. 3. The modules of the ontology of level 2 for organic chemistry

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.

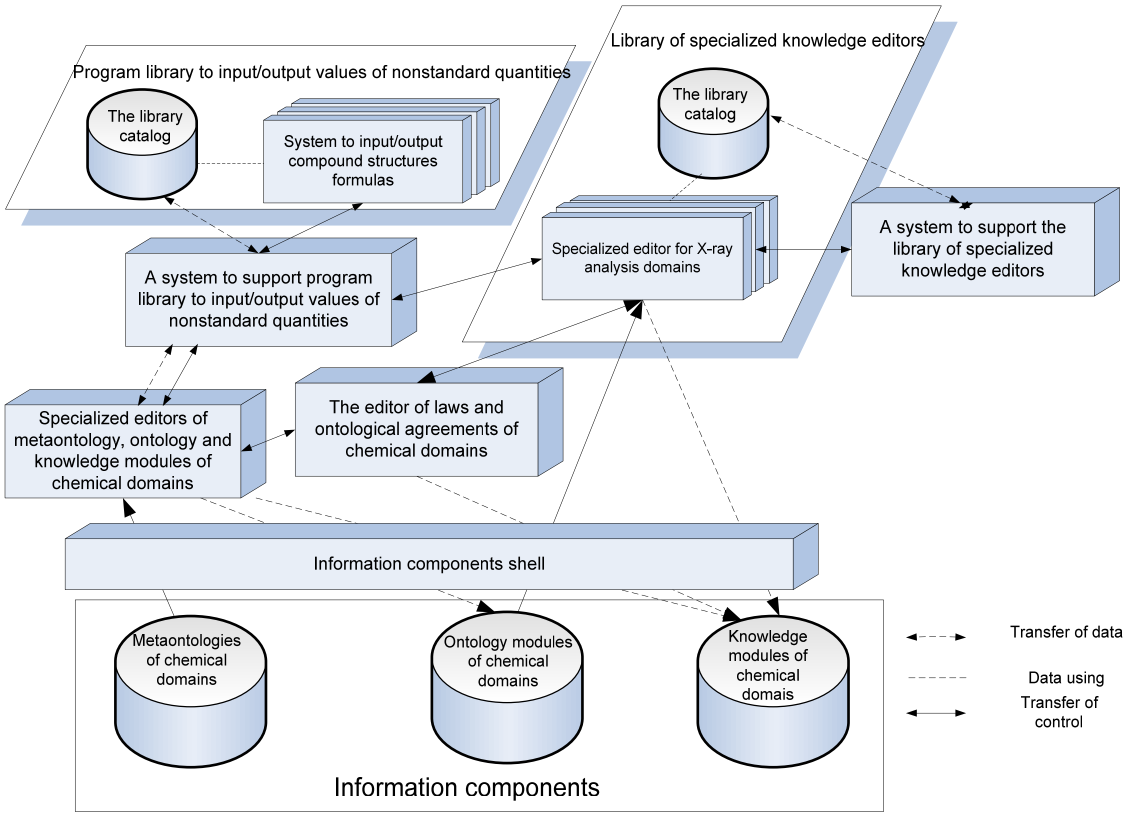

Fig. 4. Components of upgradeable editor for chemistry

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 $\equiv$ ($\cup$(n: I[1, ∞)) {(v: (($\times$ chemical elements, I[1,∞))$\Uparrow$ n)) (&(i: I[1, length(v)]) (&(j: I[1, length(v)] \ {i}) π(i, v) $\neq$ π(j, v)))}) $\cup$ ($\cup$(n: I[1, ∞)) {(v: (($\times$ chemical elements $\cup$ possible formula of substance, I[1, ∞))$\Uparrow$ n)) (&(i: I[1, length(v)]) (&(j: I[1, length(v)] \ {i}) π(i, v) $\neq$ 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 $\equiv$ (λ (v1: possible formula of substance) (v2: {(v3: chemical elements $\cup$ 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) $\cap$ j(Type2) = $\varnothing$ 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 $\neq \varnothing$, 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= <α^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 $EN^m \cup ER^m \cup EP^m \cup EC^m \ne \emptyset$.

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 $t \equiv (\lambda(v_1: M_1)\ (v_2: M_2) \ldots (v_q: M_q)\ M(v_1,\ldots v_q))$ defined at the level m to the notion name:

sort p: $t(c_1, c_2,\ldots c_{q1})$, where p is the notion name, $t(c_1, c_2,\ldots c_{q1})$ is application of the constructor of the first order defined at the level m, where $c_1 \in J_{\alpha,\theta 1}(M_1)$, $c_2 \in J_{\alpha,\theta 2}(M_2)$, …, $c_q \in J_{\alpha,\theta q}(M_q)$, where $\alpha = \alpha^m_P$, θ_1 is empty substitution $\theta_2 =\{v_1/c_1\},\ldots,\theta_q=\{v_1/c_1,\ldots,v_{q-1}/c_{q-1}\}$.

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

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 $t \equiv (\lambda(v^1_1: M^1_1)$ $(v^1_2: M^1_2) \ldots (v^1_{q1}: M^1_{q1})\ (\lambda(v^2_1: M^2_1(v_1,\ldots v_{q1}))\ (v^2_2: M^2_2(v_1,\ldots v_{q1})) \ldots (v^2_{q2}: M^2_{q2}(v_1,\ldots v_{q1})) \ldots (\lambda(v^s_1: M^s_1(v_1,\ldots v_{qs-1}))$ $(v^s_2: M^s_2(v_1,\ldots v_{qs-1})) \ldots (v^s_{qs}: M^s_{qs}(v_1,\ldots v_{qs-1}))\ M(v^1_1,\ldots v^s_{qs}))\ldots))$ defined at the level m m: $t1 \equiv t(c_1, c_2,\ldots c_{q1})$, where

- - t_1 is constructor name;
- - $c_1 \in J_{\alpha,\theta 1}(M^1_1)$, $c_2 \in J_{\alpha,\theta 2}(M^1_2)$, …, $c_{q1} \in J_{\alpha,\theta q1}(M^1_{q1})$, где $\alpha = \alpha^m_P$, θ_1 is empty subsitution, $\theta_2 =\{v_1/c_1\},\ldots,\theta_{q1}=\{v_1/c_1,\ldots,v_{q1-1}/c_{q1-1}\}$;

- - $t(c_1, c_2,\ldots c_{q1})$ 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 ≡ Own properties of entities(Chemical elements)
- Own properties of substances ≡ Own properties of entities(Chemical substances)
- Own properties of organic compounds ≡ 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 $P^m \neq \emptyset$, $\alpha^m{}_P \neq \emptyset$;

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

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

5. $EN^m \cup EC^m \neq \emptyset$;

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).

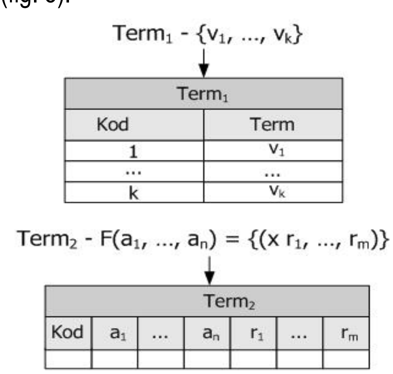

Fig 5. Relation between an ontology term and structure of a database table

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.

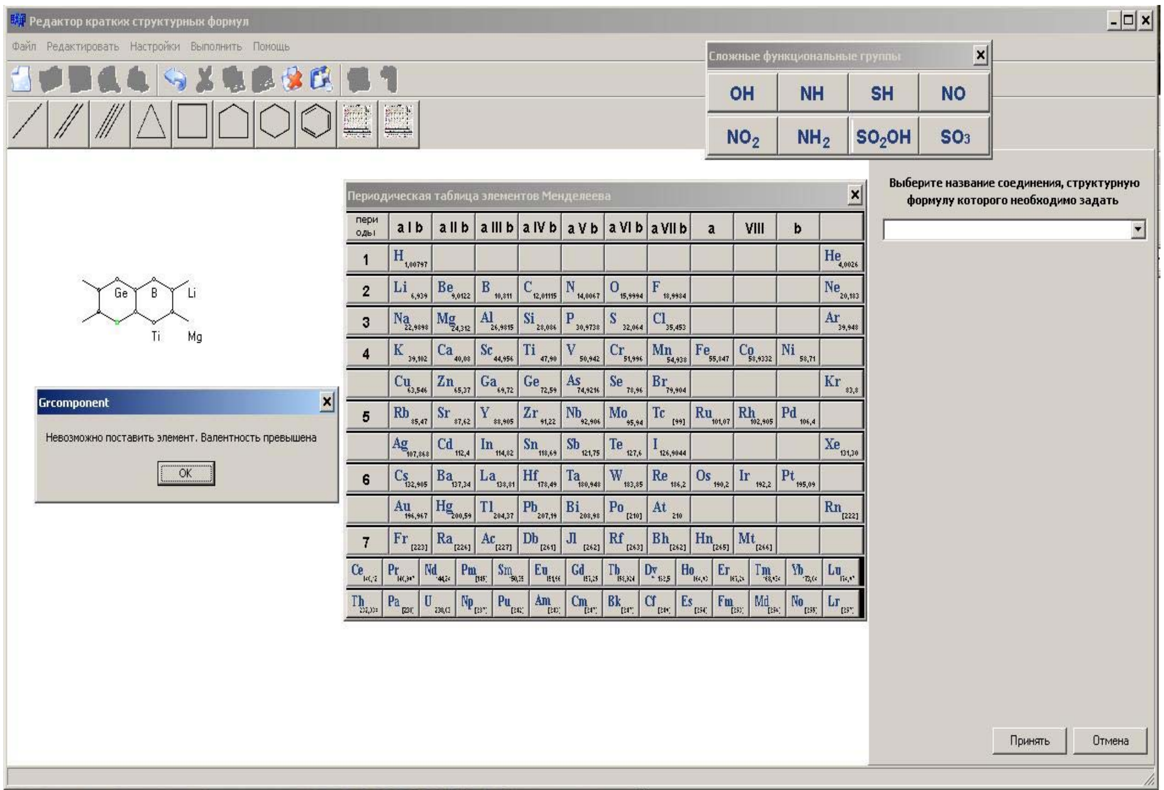

Fig. 6. An interface of the specialized graphical editor for structured formulas

The ontology defines the set of possible types of bond: bond types ≡ {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 ≡ I[1, ∞).

The set of mutual relations between the elements is represented with the term “set of bonds” ≡ (∪ (n:I[1, maximum number of bonds])(× chemical elements, element numbers, bond types)⇑ 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 ≡ (×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 ≡ {(f: {(n: I[1,∞)) possible components of the structured formula⇑ n})(&(i: I[1, length(f)])(&(j1: {(j: I[1, length(f)]) i ≠ j}) π(2, π(i, f)) ≠ 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