# UPGRADABLE TREE LEVELS EDITOR OF METAONTOLOGIES, ONTOLOGIES AND KNOWLEDGE FOR A CHEMISTRY[1]

Irene Artemieva, Natalia Reshtanenko, Vadim Tsvetnikov

*Abstract: Development of upgradable tree levels editor of metaontologies, ontologies and knowledge for a chemistry intellectual system is described. A fragment of chemistry ontology of the third level 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 multi-levels editor of ontologies and knowledge, domain ontology, chemistry ontology, domain knowledge*

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

*Conference: The paper is selected from XIV$^{th}$ International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

## Introduction

While solving applied chemical tasks researchers have to use ontologies and knowledge of different chemical domains and in turn to solve nested tasks (as the 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 3-level chemical ontology [Artemieva, 2007]. The upper level – called chemical metaontology – describes the structure of several 2-nd level chemical ontologies, 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.

There are editors [Corcho et al, 2003], [Denny, 2002] 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

[1] This paper has supported by Far Eastern Branch of the Russian Academy of Sciences, the project 06-III-A-01-005 and Russian Fund of Fundamental Investigation, the project 06-07-89071-a

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 3-level chemical ontology [Artemieva, 2007].

## Database structure for storing ontologies

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 by 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 \ ∅.

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 ∪ I ∪ 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 → {}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. 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. 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.

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. Multilevel ontologies for domains with complicated structures // In the Proceedings of the XIII-th International Conference "Knowledge-Dialog-Solution" - KDS 2007, June 18-24, Varna, Bulgaria, Sofia:FOI ITHEA.2007.Vol. 2. PP. 403-410.

[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, 2002] Denny M. Ontology Building: a Survey of Editing Tools.
URL: http://www.xml.com/pub/a/2002/11/06/ontologies.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* artemeva@iacp.dvo.ru

*Natalia V. Reshtanenko* reshtanenko@iacp.dvo.ru

*Vadim A. Tsvetnikov*

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