A Structure of Domain Ontologies and their Mathematical Models

Alexander S. Kleshchev, Irene L. Artemjeva

Expert System Department
Institute for Automation & Control Processes
Far Eastern Branch of the Russian Academy of the Sciences
5, Radio Street, Vladivostok, 690041, Russia,
Tel: 7 4232 314001, Fax: 7 4232 310452,
E-mail: kleschev@iacp.dvo.ru, artemeva@iacp.dvo.ru
page: www.iacp.dvo.ru/es/

Absract

A primitive conceptualization is defined as the set of all intended situations. A non–primitive conceptualization is defined as the set of all the pairs every of which consists of an intended knowledge system and the set of all the situations admitted by the knowledge system. The reality of a domain is considered as the set of all the situation which have ever taken place in the past, are taking place now and will take place in the future. A conceptualization is defined as precise if the set of intended situations is equal to the domain reality. The representation of various elements of a domain ontology in a model of the ontology is considered. These elements are terms for situation description and situations themselves, terms for knowledge description and knowledge systems themselves, mathematical terms and constructions, auxiliary terms and ontological agreements. It has been shown that any ontology representing a conceptualization has to be non–primitive if either (1) a conceptualization contains intended situations of different structures, or (2) a conceptualization contains concepts designated by terms for knowledge description, or (3) a conceptualization contains concept classes and determines properties of the concepts belonging to these classes, but the concepts themselves are introduced by domain knowledge, or (4) some restrictions on meanings of terms for situation description in a conceptualization depend on the meaning of terms for knowledge description.

Keywords:

Domain ontology, domain ontology model, domain model

Introduction

At present the importance of studying properties of domain ontologies is generally recognized. There are many articles devoted to this problem. But the elements and the structure of domain ontologies are not defined precisely yet as well as the representation of these elements and the structure in ontology models. The goal of this report is to make a contribution to solving this problem.

Existing Approaches to Defining the Notion of a Domain Ontology (Background)

At present three main approaches to defining the notion of a domain ontology and the notions of knowledge and a conceptualization associated with it can be recognized.

The first one, arbitrarily called here as humanitarian, suggests definitions in terms understood intuitively. Some examples of these definitions can be find in the articles [1–7]. The central merit of the humanitarian approach is the attempts made within its framework to clarify a meaningful essence of the notion of a domain ontology and other ones associated with it. The key demerit of all these definitions and the whole humanitarian approach is that a technical notion necessary for solving technical problems cannot be defined in such a manner.

The second approach to defining the notion of a domain ontology can be arbitrary called the computer one. Within the framework of the approach computer languages for domain ontology representation have been developed. Some examples of computer languages for domain ontology representation are given in [8–11].

The third approach to defining the notion of a domain ontology can be arbitrary called the mathematical one. Within the framework of this approach some attempts are made to define the notion of domain ontology in mathematical terms, or by mathematical constructions [12–14]. By and large the definitions of the mathematical approach offer considerable advantages over the definition of the computer approach, both for the fact that they have less number of technical details at the same level of rigor, and for explicit specialization in formalizing the notion of a domain ontology. The central flaw, crucial for the development of the mathematical approach, is absence of any explicit assumptions about properties of domains, their ontologies, conceptualizations and knowledge (which are characteristic of the humanitarian approach), and explicit association between these assumptions and some elements of mathematical models.

Thus, it is possible to consider that till now no generally accepted definition of domain ontology has been suggested [5,6,15,16]. However, it is possible to select different meaning of the term of domain ontology from the overview (that has been considered in the article [17]). (1) A domain ontology is the part of domain knowledge that is not to be changed. The other part of domain knowledge may be changed according to the domain ontology. (2) A domain ontology is that part of domain knowledge that restricts the meanings of domain terms. These meanings do not depend on the other part of the domain knowledge. (3) A domain ontology is a set of agreements about the domain. The other part of the domain knowledge is a set of empirical or other laws. The ontology determines a degree of consensus among domain specialists of the domain terms meaning. (4) A domain ontology is an external approximation represented explicitly of a conceptualization given implicitly. The conceptualization is a subset of the set of all the situations that can be represented. The set of situations corresponding to a knowledge base is a subset of the conceptualization. This subset is an approximation of the set of the situations possible in the reality.

All these meanings of the notion of domain ontology supplement each other.

In what follows an attempt will be made to give another definition of the notion of domain ontology. But here some basic methodological principles of this definition should be formulated. (1).On the meaningful level a domain ontology will be understood as a set of agreements (domain term definitions, their commentary, statements restricting a possible meaning of these terms, and also a commentary of these statements). These agreements are a result of understanding among members of the domain community. So, they cannot be disproved by any empirical observations. In their meaning they differ from empirical or other knowledge that can be disproved by empirical observations or in a different way. In this regard the notion of a domain ontology is analogous to the notion of a paradigm by T.S.Kuhn [18]. (2). Such properties of a domain as its ontology, conceptualization, knowledge and reality should be modeled by different parts of the same mathematical construction. (3). An explicit correspondence between these properties of domains and elements of these mathematical constructions should be defined. (4). A domain ontology model should contain both formal elements and their meaningful comments in terms understandable for specialists in the domain. (5). An ontology and its model should be compact even for complex domains containing many concepts.

A Primitive Conceptualization and Ontology of a Domain

We will consider that a professional activity is a characteristic of a domain. This activity consists in solving different tasks. Task solving needs professional knowledge, the same for all the tasks. The professional knowledge and also input and output data for every task can be represented verbally. A domain is considered as a set of the tasks, which are solved by specialists of this domain. When solving a task, a person uses a finite set of objects and relations among them. This finite set will be called a ***situation*** (a state of affairs in terms of [14]). Contrary to the definition given in [14], we will suggests that different situation can contain different sets of objects. The set of all possible situations in a domain which have ever taken place in the past, are taking place now and will take place in the future will be called the **reality** of the domain. Thus, the reality is an infinite set of separated situations [19,20]. The reality has the property that the persons studying the domain do not know the reality completely. Only a finite subset of situations forming the reality and having taken place in the past is known (although the information forming these situations also can be not completely known).

When solving tasks of a domain, a person uses an idea about the domain. This idea can be represented as an implicitly given set of intended situations called a primitive conceptualization of the domain. Every adequate conceptualization contains the reality of the domain [14]. Therefore, a primitive conceptualization is an external approximation of the reality. To represent a conceptualization it is necessary to have an appropriate concept system. A concept system is a set of definitions for basic and auxiliary concepts. To define a concept system means to give terms for concept designations, to define the capacity of every concept, and to define relationships among the terms. Auxiliary terms are introduced to make a concept system description more compact. A value of an auxiliary term is defined by the values of main and other auxiliary domain terms. Mathematical terms have universally accepted interpretation but can belong to the set of domain concepts. Therefore, these terms should be defined outside of the concept system, in mathematics. Different domains can require different mathematical terms.

One of the means for building concept system models is the language of applied logic [21]. The language of applied logic determines mathematical terms and constructions used for concept system descriptions. The kernel of the applied logic language determines a minimal set of logical means. The standard extension of the language apart from additional logical means introduces arithmetic and set-theoretic constants, operations and relations. Every specialized extension of the language gives us a possibility to define both additional logical means and constants, operations and relations of other divisions of mathematics, i.e. additional mathematical terms.

The terms of the kernel of applied logic are:

1. a name n;
2. a variable v;
3. N and L;
4. $t_1 \rightarrow t_2$, where t_1 and t_2 are terms;
5. $(\times\ t_1, t_2, ..., t_k)$, where $t_1,...,t_k$ are terms;
6. $t(t_1,...,t_k)$, where t , $t_1,...,t_k$ are terms;
7. j(t), where t is a term.

The formulas of the kernel of applied logic are:

1. $t(t_1,...,t_k)$, where t , $t_1,...,t_k$ are terms;
2. $\neg f_1$, f_1 & f_2, $f_1 \vee f_2$, $f_1 \Rightarrow f_2$, $f_1 \Leftrightarrow f_2$, where f_1 and f_2 are formulas.

A proposition of an applied logical theory consists of a prefix and a body. A prefix is a sequence of variable descriptions $(v_1:t_1)(v_2:t_2)...(v_m:t_m)$ (bounded universal quantifiers), where $(v_i:t_i)$ is a variable description, v_i is a variable, t_i is a term for all i=1,...,m. The term t_1 contains no free variables. For i=2,..., m only the variables v_1, $v_2,...,v_{i-1}$ can be free variables of the term t_i . A sequence of variable descriptions can be empty. All the variables v_1, $v_2,..., v_m$ are mutually different.

The body of a proposition depends on the type of the proposition. The types of propositions are a value description for a name, a sort description for a name, a restriction on the interpretation of names. Any free variable which is a part of the body of a proposition must be described in its prefix. If a variable is bound in the body of a proposition then it cannot be a part of the prefix of the proposition. The body of a value description for a name has a form $t_1 \equiv t_2$, where t_1 and t_2 are terms. The body of a sort description for a name has a form $\chi(t_1) = t_2$, where t_1 and t_2 are terms. The body of a restriction on the interpretation of names is a formula.

An applied logical theory named $T(E_1,E_2,...,E_k)$, where $E_1,E_2,...,E_k$ are the names of extensions of the language used for representing the theory, is a pair <TS, SS>, where TS is a finite set (perhaps empty) of names of other theories, SS is a finite set (perhaps empty) of propositions. Any applied logical theory T = <TS, SS> by definition is equivalent to an applied logical theory <∅, SS'>, where SS' is the result of the following process. Let us denote ts(T) = TS, ss(T) =SS. Let TS_1= ts(T) and SS_1= ss(T). For every i = 1,2,... let $TS_{i+1} = \bigcup_{t\in TS_i} ts(t)$, $SS_{i+1} = SS_i \cup \bigcup_{t\in TS_i} ss(t)$. If $TS_n = \varnothing$ on a recurrent step n then SS' = SS_n. The theory <∅, SS'> will be called the reduction of the theory <TS, SS>.

Semantics of terms and formulas determines the values of terms and formulas and also the conditions under which these values exist. In this case it is suggested that a function α is given on the set of names. For every name the value of the function is an interpretation of the name. The values of terms and formulas will be defined in relation to an interpretation function α and an arbitrary admissible substitution θ of values for all the free variables in the term or in the formula. If a variable being free in a term or in a formula is also free in the proposition including the term or the formula then in an admissible substitution θ the value for the variable is determined by the semantics of the proposition. But if a variable being free in a term or in a formula is bound in the proposition including the term or the formula then in an admissible substitution θ the value for the variable is determined by the semantics of the term or of the formula in that the variable is bound. Let $J_{\alpha,\theta}(t)$ denote the value of a term t for an interpretation function α and an admissible substitution θ, $J_{\alpha,\theta}(f)$ denote the value of a formula f for an interpretation function α and an admissible θ, $\theta(v)$ denote the value of a variable v in the substitution θ.

The values of terms are defined by the following way.

1. $J_{\alpha,\theta}(n) = \alpha(n)$, where n is a name; $J_{\alpha,\theta}(n)$ does not depend on θ; the value $J_{\alpha,\theta}(n)$ exists if n is an element of the set $J_{\alpha,\theta}(N)$;
2. $J_{\alpha,\theta}(v) = \theta(v)$, where v is a variable;
3. $J_{\alpha,\theta}(N)$ is the infinite set of all possible names; $J_{\alpha,\theta}(N)$ does not contain all the names that are described in the standard and in any used specialized extension of the language and also "N", "L", "≡","=", "→", "×", "⇒", "∨", "&", "¬", "⇔", "(", ")", ":", "true", "false", ",", "χ", "j"; $J_{\alpha,\theta}(N)$ does not depend on α and θ;
4. $J_{\alpha,\theta}(L)$ is the set consisting of two elements true and false; $J_{\alpha,\theta}(L)$ does not depend on α and θ;
5. $J_{\alpha,\theta}(t_1 \rightarrow t_2)$ is the set of all possible completely defined functions from the set $J_{\alpha,\theta}(t_1)$ to the set $J_{\alpha,\theta}(t_2)$; the value of the term exists if the both values $J_{\alpha,\theta}(t_1)$ and $J_{\alpha,\theta}(t_2)$ are sets;
6. $J_{\alpha,\theta}(\times\ t_1, t_2, ..., t_k)$ is the Cartesian product of the sets $J_{\alpha,\theta}(t_1), J_{\alpha,\theta}(t_2), ..., J_{\alpha,\theta}(t_k)$; the value of the term exists if all the values $J_{\alpha,\theta}(t_1), J_{\alpha,\theta}(t_2), ..., J_{\alpha,\theta}(t_k)$ are sets; the operation "×" has all the properties of Cartesian product but associativity $J_{\alpha,\theta}(\times(\times\ t_1, t_2), t_3) \neq J_{\alpha,\theta}(\times\ t_1, (\times\ t_2, t_3))$;
7. $J_{\alpha,\theta}(t(t_1,t_2,...,t_k)) = \varphi(J_{\alpha,\theta}(t_1), J_{\alpha,\theta}(t_2),..., J_{\alpha,\theta}(t_k))$ is the value of the function φ which is the interpretation of the name $J_{\alpha,\theta}(t)$ (i.e. $\varphi = \alpha(J_{\alpha,\theta}(t))$), applied to the arguments $J_{\alpha,\theta}(t_1),..., J_{\alpha,\theta}(t_k)$; the value of the term exists if the value $J_{\alpha,\theta}(t)$ is a name, having a sort (s'→ s), where s' is the Cartesian product of the sets $s_1, s_2,...$, s_k or a subset of the Cartesian product, s is a set, with $s \neq J_{\alpha,\theta}(L)$, $<J_{\alpha,\theta}(t_1), J_{\alpha,\theta}(t_2),..., J_{\alpha,\theta}(t_k)> \in s'$; in this case $J_{\alpha,\theta}(t(t_1,t_2,...,t_k)) \in s$; let us notice that if t' is such a

term that $J_{\alpha,\theta}(t') = \langle J_{\alpha,\theta}(t_1), J_{\alpha,\theta}(t_2),..., J_{\alpha,\theta}(t_k)\rangle$ then $J_{\alpha,\theta}(t(t')) = J_{\alpha,\theta}(t(t_1,t_2,...,t_k))$;

8. $J_{\alpha,\theta}(j(t)) = \alpha(J_{\alpha,\theta}(t))$ is the interpretation of the name $J_{\alpha,\theta}(t)$; the value of the term exists if $J_{\alpha,\theta}(t)$ is a name.

The values of formulas are defined in the following way.

1. $J_{\alpha,\theta}(t(t_1,...,t_k)) \Leftrightarrow \rho(J_{\alpha,\theta}(t_1), J_{\alpha,\theta}(t_2),..., J_{\alpha,\theta}(t_k))$ is the value of the predicate ρ, which is the interpretation of the name $J_{\alpha,\theta}(t)$ (i.e. $\rho = \alpha(J_{\alpha,\theta}(t))$) applied to the arguments $J_{\alpha,\theta}(t_1),..., J_{\alpha,\theta}(t_k)$; the formula has a value if the value $J_{\alpha,\theta}(t)$ is a name having a sort $(s' \to L)$, where s' is the Cartesian product of the sets $s_1, s_2,..., s_k$ or a subset of the Cartesian product, $\langle J_{\alpha,\theta}(t_1), J_{\alpha,\theta}(t_2),..., J_{\alpha,\theta}(t_k)\rangle \in s'$; let us notice that if t' is such a term that $J_{\alpha,\theta}(t') = \langle J_{\alpha,\theta}(t_1), J_{\alpha,\theta}(t_2),..., J_{\alpha,\theta}(t_k)\rangle$ then $J_{\alpha,\theta}(t(t')) \Leftrightarrow J_{\alpha,\theta}(t(t_1,t_2,...,t_k))$;

2. $J_{\alpha,\theta}(\neg f_1) \Leftrightarrow \neg J_{\alpha,\theta}(f_1)$, i.e. the value of the formula $\neg f_1$ is true if and only if the value $J_{\alpha,\theta}(f_1)$ is false; the formula has a value if the formula f_1 has a value for the interpretation function α and the substitution θ;

3. $J_{\alpha,\theta}(f_1 \ \& \ f_2) \Leftrightarrow J_{\alpha,\theta}(f_1) \ \& \ J_{\alpha,\theta}(f_2)$, i.e. the value of the formula $f_1 \ \& \ f_2$ is true if and only if the both values $J_{\alpha,\theta}(f_1)$ and $J_{\alpha,\theta}(f_2)$ are true; the formula has a value if the both formulas f_1 and f_2 have values for the interpretation function α and the substitution θ;

4. $J_{\alpha,\theta}(f_1 \vee f_2) \Leftrightarrow J_{\alpha,\theta}(f_1) \vee J_{\alpha,\theta}(f_2)$, i.e. the value of the formula $f_1 \vee f_2$ is true if and only if at least one of the values $J_{\alpha,\theta}(f_1)$ or $J_{\alpha,\theta}(f_2)$ is true; the formula has a value if the both formulas f_1 and f_2 have values for the interpretation function α and the substitution θ;

5. $J_{\alpha,\theta}(f_1 \Rightarrow f_2) \Leftrightarrow J_{\alpha,\theta}(f_1) \Rightarrow J_{\alpha,\theta}(f_2)$, i.e. the value of the formula $f_1 \Rightarrow f_2$ is true if and only if either the value $J_{\alpha,\theta}(f_1)$ is false or the both values $J_{\alpha,\theta}(f_1)$ and $J_{\alpha,\theta}(f_2)$ are true; the formula has a value if the both formulas f_1 and f_2 have values for the interpretation function α and the substitution θ;

6. $J_{\alpha,\theta}(f_1 \Leftrightarrow f_2) \Leftrightarrow J_{\alpha,\theta}(f_1) \Leftrightarrow J_{\alpha,\theta}(f_2)$, i.e. the value of the formula $f_1 \Leftrightarrow f_2$ is true if and only if either the both values $J_{\alpha,\theta}(f_1)$ and $J_{\alpha,\theta}(f_2)$ are false or the both values $J_{\alpha,\theta}(f_1)$ and $J_{\alpha,\theta}(f_2)$ are true; the formula has a value if the both formulas f_1 and f_2 have values for the interpretation function α and the substitution θ;

Semantics of propositions determines the meaning of the propositions and also the conditions under which propositions have meaning.

The set of admissible substitutions θ for free variables of a proposition is formed in the following way. If the prefix of the proposition is empty then the set of admissible substitutions of the proposition consists of the only empty substitution. Let the prefix of the proposition be of the form $(v_1:t_1)(v_2:t_2)...(v_m:t_m)$, then the set of admissible substitutions is the set of all the substitutions of the form $\theta=(v_1/c_1,...,v_m/c_m)$, where $c_1 \in J_{\alpha\theta_1}(t_1)$, $c_2 \in J_{\alpha\theta_2}(t_2)$, ..., $c_m \in J_{\alpha\theta_m}(t_m)$, θ_1 is the empty substitution, $\theta_2 = (v_1/c_1)$,..., $\theta_m=(v_1/c_1,...,v_{m-1}/c_{m-1})$. The proposition has meaning if $J_{\alpha\theta_1}(t_1)$, $J_{\alpha\theta_2}(t_2)$,..., $J_{\alpha\theta_m}(t_m)$ are such sets that the set of admissible substitutions is finite.

A value description for a name with the body $t_1 \equiv t_2$ has the following meaning: for every admissible substitution θ the interpretation of the name $J_{\alpha,\theta}(t_1)$ is $J_{\alpha,\theta}(t_2)$. The proposition has meaning if for all the admissible substitutions the value $J_{\alpha,\theta}(t_1)$ is a name, the value of the term t_2 exists for the interpretation function α and for the substitution θ and also it does not follow from the logical theory that the name $J_{\alpha,\theta}(t_1)$ has more than one value. In addition, if the name $J_{\alpha,\theta}(t_1)$ has a sort s then for a proposition to have meaning it is necessary that the value $J_{\alpha,\theta}(t_2)$ should be an element of the set s. A set of value descriptions for names can contain recursive value definitions for names.

A sort description for a name with the body $\chi(t_1) = t_2$ has the following meaning: for every admissible substitution θ the name $J_{\alpha,\theta}(t_1)$ has the sort $J_{\alpha,\theta}(t_2)$. The proposition has meaning if for all the admissible substitutions $J_{\alpha,\theta}(t_1)$ is a name, $J_{\alpha,\theta}(t_2)$ is a set and it does not follow from the logical theory that the name $J_{\alpha,\theta}(t_1)$ has more than one sort. A set of sort descriptions for names can contain recursive sort definitions for names.

A restriction on the interpretation of names has the following meaning: an interpretation function α is admissible if $J_{\alpha\theta}(f)$ = true for all the admissible substitutions θ, where f is a formula that is the body of this proposition. The proposition has meaning if there is such an interpretation function that the formula f is true for all the admissible substitutions θ.

Now we define semantics of applied logical theories. The set of names being parts of an applied logical theory can be divided into two nonintersecting subsets: a set of uniquely interpreted names and a set of ambiguously interpreted names. A name is uniquely interpreted if one of the following conditions is met:

- the applied logical theory determines neither any sort nor any value for a name n; in this case for any α the interpretation $\alpha(n) = n$;

- the applied logical theory determines a value e for a name n and the value does not depend on the interpretations of other names; in this case for any α the interpretation $\alpha(n) = e$;

- the applied logical theory determines a value e for a name n and the value is uniquely determined by the interpretations of other names.

All the other names are ambiguously interpreted. For every such a name the applied logical theory determines a sort s but does not determine any value. In this case any interpretation function α must meet the restriction $\alpha(n) \in s$.

An interpretation function α is admissible for an applied logical theory if all the propositions of the theory reduction have meaning for this interpretation function. An applied logical theory is semantically correct if there is an admissible interpretation function α. Since for every proposition the set of admissible substitutions is determined uniquely but the admissible interpretation function is determined ambiguously then a semantically correct applied logical theory determines a set of admissible interpretation functions. It is easily seen that under these conditions the set of ambiguously interpreted names of any semantically correct applied logical theory is finite for any admissible interpretation function.

The constriction of an admissible interpretation function α to the set of ambiguously interpreted names of an applied logical theory will be called a model of the theory. A model of an applied logical theory can be represented by such a set of value descriptions for names that after adding the set to the theory all the names of the new theory built in such a way will be uniquely interpreted.

Now we consider an example of definition of a specialized extension of the applied logic language (another examples can be find in [21]).

Example 1. The specialized extension Differentiation of unary functions of the language of applied logic will be described. The term is

1. dt(v)/dv, where t(v) is a term depending on the variable v; $J\alpha\theta(dt(v)/dv)$ is the derivative of the function $J\alpha\theta(t(v))$ with respect to v; the value of the term exists if the function $J\alpha\theta(t(v))$ is a differentiable function.

The extension defines no new types of formulas.

An applied logical theory [21] is a model of a concept system. The definitions of basic terms are represented by a set of sort description for names. The definitions of auxiliary terms are represented by a set of value descriptions for names. The relations among terms are represented by a set of restrictions on the interpretation of names. The sort description for a term determines the set of value models for the term. In any (real or imaginary) situation only an element of this set can be a value of the term. Thereby, the sort description for a term determines a model of the capacity for the concept designated by the term. A model of the capacity for a concept can be either a finite or infinite set. The applied logical theory of Example 2 can be considered as an example of a concept system.

A primitive ontology of a domain determines an external approximation of the primitive conceptualization of the domain (a conceptualization considered as the set of all intended situations is a subset of the set of all the models for its ontology), since the set of ontological agreements can be incomplete. A primitive ontology of a domain is a concept system of the domain reality, i.e. the primitive ontology determines the terms of the primitive conceptualization (terms for situation description).

One of the means for building a primitive ontology model is an unenriched logical relationship system without parameters [21]. An unenriched logical relationship system O without parameters. is a semantically correct applied logical theory Φ, having at least one ambiguously interpreted name. The set of propositions of the reduction of Φ will be called the set of logical relationships.

If an unenriched logical relationship system O without parameters is considered as a primitive ontology model, then the set all models for the set Φ of logical relationships is also an external approximation of the domain conceptualization model. An unenriched logical relationship system O without parameters can be considered as a primitive ontology model of a domain, if each of its logical relationship has a meaningful interpretation that a community of the domain agrees with, and the whole system is an explicit representation of a conceptualization of the domain.

Example 2. An unenriched logical relationship system without parameters which is a model of a simplified ontology of Dynamics: $O_1 = T_1$(ST, Intervals, Differentiation of unary functions), where T_1(ST, Intervals, Differentiation of unary functions) is an applied logical theory. The unknowns of the system are bodies, time moments, mass, coordinate, and force.

The applied logical theory T_1(ST, Intervals, Differentiation of unary functions) = $<\varnothing, SS_1>$, where SS_1 consists of the following propositions.

The value descriptions for names (the definitions of auxiliary terms).

(2.1.1) speed ≡ $(\lambda(v_1$: bodies$)(\lambda(v_2$: R)dc oordinate $(v_1)(v_2)/dv_2))$

Speed is the derivative of coordinate with respect to time.

(2.1.2) acceleration ≡ $(\lambda(v_1$: bodies$)(\lambda(v_2$: R)ds peed $(v_1)(v_2)/dv_2))$

Acceleration is the derivative of speed with respect to time.

The sort descriptions for names (the definitions of terms for situation description).

(2.2.1) χ(bodies) = {}N

Bodies me ans a set of physical bodies.

(2.2.2) χ(time moments) = (bodies → {}(R[0,∞]))

Time moments means a function that takes a body and returns the set of the time moments in which the body was observed in the situation; the unit of measurement of time is s.

(2.2.3) χ(mass) = (bodies → R[0, ∞])

Mass means a function that takes a body and returns its mass; the unit of measurement of mass is kg.

(2.2.4) χ(c o o r d i n a t e) = (b o d i e s → (R → R))

Coordinate means a function that takes a body and returns a function that takes a time moment and returns the coordinate of the body at the moment; in this model the space is taken to be one-dimensional; the unit of measurement of coordinate is m.

(2.2.5) χ(f o r c e) = (b o d i e s → (R→R[0,∞]))

Force means a function that takes a body and returns a function, that takes a time moment and returns the force acting on the body at the moment; if the force is positive then the direction of the force vector and the direction of the body movement are the same; if the force is negative then the directions are opposite; the unit of measurement of force is N.

Ontological agreements about a domain are represented by a set of restrictions on the interpretation of names of the unenriched logical relationship system which is an ontology model of the domain. Ontological agreements are explicitly formulated agreements about restrictions on the meanings of the terms in which the domain is described (additional restrictions on capacity of the concepts designated by these terms). If a domain ontology model is an unenriched logical relationship system without parameters, then all the ontological agreements are only constraints of situation models. The set of ontological agreements, in this case, can be empty, too. The ontology model of example 2 is an example of an ontology model represented by an unenriched logical relationship system without parameters and with the empty set of ontological agreements.

Objects in situation models can be represented: by elementary mathematical objects (numbers and so on); by names having neither sort nor value [21] (such a name is a designation of an object); by structural mathematical objects (sets, n-tuples, and so on) constructed of elementary or structural mathematical objects or names having neither sort nor value by composition rules defined in the language of applied logic. In Example 2 the values of the speed, acceleration and mass are represented by functions whose values are real numbers; the values of the coordinate of a body and the value of the force acting on a body are represented by functions whose values are also functions; the values of time moments of body observations are represented by a function whose values are sets of real numbers; the bodies in situations are represented by names having neither sort nor value.

The set of names having neither sort nor value and used as designations of objects (and their components) in situation models can be determined explicitly or implicitly in a model of primitive ontology of a domain. In the former case, all these names appear in sort descriptions for unknowns. A domain ontology model can determine only some of the names having neither sort nor value and used in situations for designating objects. The other names are determined by a model of situation and may have different meaning in different situations. In Example 2 the names having neither sort nor value and not fixed by the domain ontology model are the designations of bodies.

Unknowns represent relations among objects depending on situations. In different situations the relations corresponding to the same unknown can be different. Every objective unknown designates a role that an (unique) object of the situation plays in each situation, and also in every situation there is its own object playing this role. In Example 2 the unknown b o d i e s represents the role that is played by a set of bodies. Every functional unknown designates a set of functional relations. For each situation this functional relation is the one among objects of the situation. For different situations these relations corresponding to the same unknown can be different. In Example 2 the unknowns t i m e   m o m e n t s, m a s s, c o o r d i n a t e and f o r c e represent functional relations between objects of a situation. Analogously, every predicative unknown designates a set of nonfunctional relations. For each situation this nonfunctional relation (it may be empty) is the one among objects of the situation. For different situations these relations corresponding to the same unknown can be different. Thus, every unknown can be considered as a designation of a one-to-one correspondence between situations and the values of the unknown in these situations. The set of the unknowns, whose values form a model of a situation, will be called the structure of the situation model.

The sort description for an unknown determines the set of value models for the unknown. In any (real or imaginary) situation only an element of this set can be a value of the unknown. Thereby, the sort description for an unknown determines a model of the capacity for the concept designated by the unknown. A model of the capacity for a concept can be both a finite and infinite set. In Example 2 propositions 2.2.1 - 2.2.5 determine models of the capacity for concepts designated by the unknowns b o d i e s, t i m e   m o m e n t s, m a s s, c o o r d i n a t e, and f o r c e. A model of a (real or imaginary) situation is a set of values of the unknowns for the unenriched logical relationship system representing a domain ontology model. A model of a situation can be represented by a set of value descriptions for all the unknowns.

Example 3. A model of a situation for Dynamics (a domain ontology model is represented in Example 2)

(3.1) b o d i e s ≡ {m a s s   p o i n t}

The only body m a s s   p o i n t is considered.

(3.2) t i m e   m o m e n t s ≡ (λ(v: {m a s s   p o i n t}) / (v = m a s s   p o i n t ⇒ {0, 5, 10})/

The motion of mass point was observed in 0, 5, and 10 s after the b e g i n n i n g of the situation.

(3.3) m a s s ≡ (λ(v: {m a s s   p o i n t}) /(v = m a s s   p o i n t ⇒ 0,002/)

The mass of mass p o i n t is 2g.

(3.4) f o r c e ≡ (λ(v: {ma ss point}) /(v = m a s s p o i n t ⇒ (λ(v_1: {0, 5, 10}) 0)/)

No force acted on mass p o i n t at any moment of its observation.

(3.5) c o o r d i n a t e ≡ (λ(v: {m a s s p o i n t })/(v = m a s s p o i n t ⇒ (λ(t: {0, 5, 10}) 5 * t)/)

The coordinate is changed according to the law x = 5t

A model of a primitive conceptualization is the Cartesian product of capacity models for concepts designated by unknowns from which the elements contradicting to constraints of situation models are excluded. To select the situations belonging to the domain reality, it is necessary to define a knowledge system for the domain. A knowledge system of a domain is a set of empirical or other laws of the domain representing additional restrictions on the meanings of terms for situation descriptions. If an unenriched logical relationship system without parameters is a model of a primitive ontology of a domain, then any of its enrichments [21] is **a model of a knowledge system** for the domain. The primitive ontology model introduces all the concepts for the description of the domain. In this case any enrichment k of the system O is a set of logical relationships – restrictions on the interpretation of unknowns. These laws represent empirical or other laws of the domain. Since this enrichment does not introduce any new names, it cannot contain any value or sort descriptions for names [21].

Example 4. A model of a knowledge system for Dynamics (a possible enrichment of the system without parameters of Example 3) is represented by a proposition (by a restriction on the interpretation of unknowns.

(4.1) (v_1: b o d i e s)(v_2: t i m e m o m e n t s (v_1)) f o r c e (v_1, v_2) = m a s s (v_1) * a c c e l e r a t i o n (v_1, v_2)

Newton's second law of motion.

A primitive ontology specifies the following restrictions on the contain of a knowledge system: the set of the propositions representing the knowledge system must be consistent; the set of the propositions representing both the knowledge system and the ontology must be consistent. A pair consisting of a primitive domain ontology model O and a knowledge system model k is a model of the domain. If a knowledge system model is given, then the domain reality model consists of all the situation models for which all the propositions of the knowledge system model and the domain ontology model are true.

We will suggest that for any primitive conceptualization of a domain the hypothesis on its adequacy is true: the reality is a subset of the set of intended situations. In view of the reality definition it is evident that this hypothesis cannot be verified. Hence, every adequate conceptualization imposes certain limitations on the notion of the reality.

Non-Primitive Conceptualization and Ontology

A relation between the set of intended situations on a domain and the set of intended knowledge systems of the domain will be called non-primitive conceptualization of the domain. A non-primitive ontology is an explicit representation of a non-primitive conceptualization. A non-primitive ontology defines both terms for situation description and terms for knowledge description, i.e. the non-primitive ontology consists of two concept systems and a correspondence between them.

One of the means for building a model of a non-primitive domain ontology is a pure unenriched logical relationship system with parameters [21]. An unenriched logical relationship systems with parameters is a pair O = <Φ, P>, where Φ is a semantically correct applied logical theory and P is a nonempty proper subset of the set of ambiguously interpreted names of the theory Φ, P is called the set of parameters. Ambiguously interpreted names of the theory Φ which do not belong to the set P, will be called unknowns of the system O.

If an unenriched logical relationship system with parameters is a domain ontology model then all the unknowns of the system are models of terms for situation description, and all the parameters of the system are models of terms for knowledge description.

Example 5. A model of a simplified ontology for medical diagnostics in which a single examination of the patient is only considered.

The value descriptions for names

(5.1.1) s e t s o f v a l u e s ≡ ({} N) ∪ ([ ]I) ∪ ([]R)

S e t s o f v a l u e s means the set of possible value ranges for all signs; these ranges can be sets of names (ranges of qualitative values), integer-valued and real-valued intervals (ranges of quantitative values).

The sort descriptions for names.

(5.2.1) χ(s i g n s) = {}N

Signs means a finite set of medical sign names.

(5.2.2) χ(d i s e a s e s) = {}N

Diseases means a finite set of disease names.

(5.2.3) χ(p o s s i b l e v a l u e s) = (s i g n s → s e t s o f v a l u e s)

P o s s i b l e v a l u e s means a function that takes a sign and returns its possible value range.

(5.2.4) χ(n o r m a l v a l u e s) = (s i g n s → s e t s o f v a l u e s)

N o r m a l v a l u e s means a function that takes a sign and returns its normal value range.

(5.2.5) χ(clinical picture) = (diseases → ({} signs))

A clinical picture is a function that takes a disease and returns a subset of the set of signs which is the clinical picture of the disease.

(5.2.6) χ(values for a sign and a disease) = ({(v: (× diseases, signs)) π(2, v) ∈ clinical picture(π(1, v))} → sets of values)

Values for a sign and a disease means a function that takes a disease and a sign from the clinical picture of the disease and returns the set of values which are possible for the sign and for the disease.

(5.2.7) χ(diagnosis) = diseases

A diagnosis is the disease which the patient is ill with; in this model diagnosis can be either a disease or healthy.

(5.2.8) (v: signs) χ(v) = possible values(v)

Every term from the set signs means the value of the sign in the patient.

The restrictions on the interpretations of names

(5.3.1) (v: signs) (normal values(v) ≠ ∅) & (normal values(v) ⊂ possible values(v))

For any sign its set of normal values is a nonempty proper subset of its set of possible values.

(5.3.2) clinical picture(healthy)=∅

Clinical picture of healthy contains no signs.

(5.3.3) (v: signs \ clinical picture(diagnosis)) j(v) ∈ normal values(v)

For every sign not belonging to the clinical picture of the disease which the patient is ill with, the value of the sign can be only normal.

(5.3.4) (v: clinical picture(diagnosis)) j(v) ∈ values for a sign and a disease(diagnosis, v)

For every sign from the clinical picture of the disease which the patient is ill with, the value of the sign is possible for the sign and for the disease.

The set of parameters P_6 = {signs, diseases, possible values, normal values, clinical picture, values for a sign and a disease}. The unknowns are diagnosis and also the names being elements of the set of names which is the interpretation of the parameter signs.

If a domain ontology model is an unenriched logical relationship system with parameters then the set of ontological agreements can be divided into three nonintersecting groups: constraints of situation models, i.e. the agreements restricting the meanings of terms for situation description; constraints of knowledge models, i.e. the agreements restricting the meanings of terms for knowledge description; agreements setting up a correspondence between models of knowledge and situations, i.e. the agreements setting up a correspondence between the meanings of terms for situation and knowledge description. Every proposition of the first group must contain at least one unknown or a variable whose values are unknowns and cannot contain any parameters; every proposition of the second group must contain at least one parameter or a variable whose values are parameters and cannot contain any unknowns; every proposition of the third group must contain at least one parameter or a variable whose values are parameters and at least one unknown or a variable whose values are unknowns. In doing so, the definitions of auxiliary terms should be taken into account. In Example 5 propositions 5.2.1–5.2.6 define the terms for knowledge description, propositions 5.2.7–5.2.8 define the terms for situation description, the set of constraints on situation models is empty; the set of constraints on knowledge models consists of propositions 5.3.1 – 5.3.2; the set of agreements setting up the correspondence between models of knowledge and situations consists of propositions 5.3.3 – 5.3.4.

If a model of a domain ontology is an unenriched logical relationship system with parameters, then the parameters of the system are the domain terms which are used for knowledge description. If a model of domain ontology is unenriched logical relationship system O with parameters then any enrichment k of the system O is a set α_P of the parameter values for the system O [21]. A value of an objective parameter determines a feature of the domain, a set of names for situation description, or a set of parameter names. Every enrichment (a knowledge base) can introduce new names as compared with the ontology – terms for situation and knowledge description. Functional and predicative parameters represent empirical or other laws of the domain. The value of every functional or predicative parameter is some relation among terms and/or domain constants. In this case domain knowledge is described at a higher level of abstraction than in the case of primitive domain ontology. The values of parameters can be represented by a set of propositions which are value descriptions for names.

The sort description for a parameter determines the set of value models for the parameter. In any knowledge model only an element of this set can be a value of the parameter. Thereby, the sort description for a parameter determines a model of the capacity for the concept designated by the parameter. A model of the capacity for a concept can be both a finite and infinite set.

A model of a non-primitive conceptualization is a relation between the set of all the intended situation models and the set of all the intended knowledge system models. If an ontology model is a pure unenriched logical relationship system with parameters O, then a model of a set of intended knowledge systems of the domain is the set of all the enrichments of the system O [21]. This set of enrichments is the Cartesian product of capacity models for concepts designated by parameters from which the elements contradicting to constraints on knowledge models are excluded. A model of a set of intended situations is the Cartesian product of capacity models for concepts designated by unknowns from which the elements contradicting to constraints on situation models are excluded. A model of a non-primitive conceptualization is the Cartesian product of the set of intended situation models by the set of intended knowledge system models from which the tuples contradicting to constraints on correspondence between these two concept system models are excluded.

A domain ontology will be called precise if the approximation of the conceptualization determined by the ontology is precise. A conceptualization will be called precise if it is the same as the domain reality. It is apparent that the domains related to the real world have no precise conceptualization. But conceptualizations are possible for theoretical (imaginary) domains (mathematics, theoretical mechanics, theoretical physics and so on), for which their precision is postulated.

A Comparison between Different Ontology Classes

Now let us discuss the question about capabilities of primitive and non-primitive domain ontologies. Any ontology represents a non-primitive conceptualization (1) if the conceptualization contains intended situations of different structures; (2) if the conceptualization contains concepts designated by terms for knowledge description; (3) if the conceptualization contains concept classes and determines properties of the concepts belonging to these classes, and concepts themselves are introduced by domain knowledge; (4) if in the conceptualization some restrictions on meanings of terms for situation description depend on the meaning of terms for knowledge description.

The more compactly and clearly a domain ontology describes agreements about domains, the better the conceptualization represented by this ontology is. In this regard primitive ontologies require for every term for situation description to appear explicitly in these agreements. For real domains (such as medicine) their ontologies turn out immense because of large number of these terms. At the same time, the non-primitive ontologies describe agreements about domains for groups of terms, rather then only for isolated terms through using terms for knowledge description. In doing so the majority of the terms for situation description and some terms for knowledge description do not appear explicitly in agreement descriptions. As a result, these agreements become compact and more general.

The more understandable knowledge bases represented in terms of an ontology are for domain specialists, the better the domain ontology is. In this respect primitive ontologies represent knowledge bases as sets of complex propositions represented formally by logical formulas. The more complex these formulas are, the more difficult it is to understand them. At the same time, the non-primitive ontologies introduce special terms for knowledge description. The meanings of these terms are determined by ontological agreements, and their connection with terms for situation description among them. In real domains these terms are commonly used to ease mutual understanding and to make communication among domain specialists economical. The meanings of these terms are, as a rule, understood equally by all domain specialists. The role of these terms is to represent domain knowledge as a set of relation (of simple facts). It is considerably easier for domain specialists to understand the meanings of these simple facts than the meanings of arbitrary complex propositions.

The more precise approximation of a conceptualization a domain ontology assumes, the better it is. First, let us remark that it follows from the theorem about eliminating parameters of enriched logical relationship systems [21] that if there is a domain model represented by an enriched logical relationship system with parameters which determines an approximation of the domain reality, then there is a model of the domain represented by an enriched logical relationship system without parameters which determines the same approximation of the domain reality. In this regard domain models represented by enriched logical relationship systems with parameters offer no advantages over domain models represented by enriched systems without parameters.

As for domain ontology models, every one represented by an unenriched logical relationship system determines some approximations for both the set of intended domain situation models and for the set of intended domain knowledge models. If a model O_P of a domain non-primitive ontology represented by an unenriched logical relationship system with parameters determines an approximation $\bigcup_{k\in En(O_P)} A(<O_P,k>)$ of the set of intended domain situation models, then the unenriched logical relationship system O_X without parameters quasiequivalent to O_P determines the approximation $\bigcup_{k\in En(O_X)} A(<O_X,k>)$ of the same set of intended situation models [21]. Here $<O_P, k>$ is a domain model with a knowledge base k, $A(<O_P, k>)$ is the domain reality model, $En(O_P)$ is the set of all intended knowledge bases. Let $h : En(O_P) \rightarrow En(O_X)$ be the map defined by the theorem about eliminating parameters of unenriched logical relationship systems and $H = \{h(k) \mid k \in En(O_P)\}$. Then $\bigcup_{k\in En(O_X)} A(<O_X,k>) =$

$\bigcup_{k\in En(O_P)} A(<O_X,h(k)>) \cup \bigcup_{k\in En(O_X)\backslash H} A(<O_X,k>)$; but by the theorem about eliminating parameters of enriched logical relationship systems $\bigcup_{k\in En(O_P)} A(<O_X,h(k)>) = \bigcup_{k\in En(O_P)} A(<O_P,k>)$, i.e. $\bigcup_{k\in En(O_X)} A(<O_X,k>) = \bigcup_{k\in En(O_P)} A(<O_P,k>) \cup \bigcup_{k\in En(O_X)\backslash H} A(<O_X,k>)$. Thus, the approximation of the set of intended situation models determined by the system O_X, is less precise than the approximation represented by the system O_P.

If a model O_P of a domain ontology represented by an unenriched logical relationship system with parameters determines an approximation $En(O_P)$ of the set of intended domain knowledge models, then the unenriched logical relationship system O_X without parameters determines an approximation $En(O_X)$ of the same set of intended knowledge models. In this case H is a subset of $En(O_X)$, i.e. the approximation of the set of intended knowledge models determined by the system O_X also is less precise than the approximation determined by the system O_P. In what follows we show some reasons of this fact.

Let us consider the case when a domain non-primitive ontology model is a pure unenriched logical relationship system O_P with parameters. First, the constraints of knowledge models represented by O_P determine the set $En(O_P)$ as a proper subset of the set of all possible interpretations of the system O_P's parameters, whereas, if the system O_X without parameters is a domain primitive ontology model, then this ontology model contains practically no restrictions on the set $En(O_X)$. Second, for the theorem about eliminating parameters of enriched logical relationship systems a set of formulas representing empirical and other domain laws can be deduced from every proposition setting up a correspondence between knowledge models and situation models and from parameter values. These formulas contain no parameters. It is obvious that the forms of these formulas are restricted and determined by the forms of propositions setting up a correspondence between knowledge models and situation models. At the same time, if a domain primitive ontology model is an unenriched logical relationship system O_X without parameters, then this system imposes no restrictions on the form of formulas entering its enrichments.

Domain non-primitive ontologies are thus seen to offer certain advantages over domain primitive ontologies usually considered in literature (see also [17,22]).

Acknowledgments

This work was carried out with financial support of the Russian Fund of Fundamental Investigations (grant 99-01-00634).

References

[1]. Alberts L.K. 1993. YMIR: an Ontology for Engineering Design. University of Twente.

[2]. van Heijst G., Schreiber A.T., and Wielinga B.J. 1996. Using Explicit Ontologies in KBS Development. *The International Journal of Human and Computer Studies* 46 (2-3): 183-292.

[3]. Gruber, T.R. 1994. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *The International Journal of Human-Computer Studies* 43 (5/6): 907-928.

[4]. Mizoguchi R. A Step Towards Ontological Engineering. http://www.ei.sanken.osaka-u.ac.jp/english/step-onteng.html.

[5]. Studer R., Benjamins V.R., Fensel D. 1998. Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering* 25: 161 –197.

[6]. Uschold M. 1998. Knowledge Level Modeling: Concepts and Terminology. *The Knowledge Engineering Review* 13(1): 5-29.

[7]. Wielinga B.J. & Schreiber A.T. 1993. Reusable and Sharable Knowledge Bases: a European Perspective. In Proceedings of International Conference on Building and Sharing of Very Large-Scaled Knowledge Bases, 103-115. Tokyo: Japan Information Processing Development Center.

[8]. Genesereth, M.R., Fikes R.E. et al. 1992. Knowledge Interchange Format (version 3.0) Reference Manual. Interlingua Working Group of the DARPA Knowledge Sharing Effort. Technical Report Logic-92-1. Dept. of Computer Science, Stanford University.

[9]. Gruber T.R. 1992. Ontolingua: A Mechanism to Support Portable Ontologies. Technical Report KSL-91-66. *Knowledge* Systems Laboratory, Stanford University.

[10]. Gruber T.R. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5: 199-220.

[11]. Schreiber A.T., Wielinga B.J. Akkermans J.M., Van de Velde, Anjewierden A. 1995. CML: The Common KADS Conceptual Modeling Language. In Stells L. Schreiber A.T., Van de Velde (editors), A Future for Knowledge Acquisition. Proceedings of the 8th European Knowledge Acquisition Workshop EKAW'94, 1-25. Berlin/Heidelberg. Springer-Verlag.

[12]. Guarino N. 1996. Understanding, Building, and Using Ontologies. *The International Journal of Human-Computer Studies* 46(2-3).

[13]. Wielinga, B., Schreiber A.T., Jansweijer W., Anjewierden A. and van Harmelen F. 1994. Framework and Formalism for Expressing Ontologies

(Version 1). ESPRIT Project 8145 KACTUS, Free University of Amsterdam Deliverable, DO1b.1.

[14]. Guarino N. 1998. Formal Ontology and Information Systems. In Proceeding of International Conference on Formal Ontology in Information Systems (FOIS'98), 3-15. N. Guarino (ed.), Trento, Italy, June 6-8, 1998. Amsterdam, IOS Press.

[15]. Mark B. Ontologies as the Representation (and Re-Representation) of Agreement. Http://www-ksl.stanford.edu/ KR96/BillMarkPositionPaper.html

[16]. Peter S. and McGonigal D. A. 1998. Practical Approach to Achieve Knowledge Sharing and Reuse. In Appl. of Advanced Information Technologies: Proceeding of the Forth World Congress on Expert systems, vol. 1, 434-441. Mexico City Cognizant Communication Corporation.

[17]. Kleshchev A.S., Artemjeva I.L. 2000a. Mathematical Models of Domain Ontologies, Technical Report 18-2000, Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences, Vladivostok. (see also http://www.iacp.dvo.ru/es/)

[18]. Kuhn T.S. 1970. The Structure of Scientific Revolutions. Chicago.

[19]. Artemjeva I.L., Gavrilova T.L., Kleshchev A.S. 1995. Models of the Domains with Elementary Objects for Expert Systems. *Scientific-technical Information* 12: 8-18 (in Russian).

[20]. Artemjeva I.L., Gavrilova T.L., Kleshchev A.S. 1997. Logical Models of the Second Order for the Domains. Scientific-*technical Information* 6: 14-30 (in Russian).

[21]. Kleshchev A.S., Artemjeva I.L. 2000b. Unenriched Logical Relationship Systems, Technical Report 1-2000, Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences, Vladivostok. (see also http://www.iacp.dvo.ru/es/)

[22]. Kleshchev A.S., Artemjeva I.L. 1999. Domain Ontologies and Knowledge Processing. Technical Report 7-99, Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences, Vladivostok. (see also http://www.iacp.dvo.ru/es/)