The Ontology Model for Automating the Solution of Multidisciplinary Research Tasks

Alexandr F. Berman [1], Olga A. Nikolaychuk [1,2] and Alexandr I. Pavlov [1]

[1]Matrosov Institute for Systems Dynamics and Control Theory of Siberian Branch of the Russian Academy of Sciences,
Lermontov str., 134
Irkutsk, Russia
E-mail: berman@icc.ru

[2]Irkutsk State University,
Karl Marx str., 1
Irkutsk, Russia
E-mail: nikoly@icc.ru

Abstract

The proposed ontology model is instrumental in describing multidisciplinary information on the problem under consideration, including methods and means of its solution. This model is based on a hierarchy of parts of the subject and problem areas, ontology of representation and ontology of the upper level. The proposed model provides the following: representation of heterogeneous, in terms of formalization degree, information on the problem addressed; independence of the presentation of data and knowledge of their processing methods; shared development of information space.

Keywords: Domain ontology, problem ontology, top-level ontology, representation ontology, application ontology, multi-disciplinarity.

1. Introduction

The existing level of theoretical bases of creation and support of problem-oriented software systems based on knowledge does not ensure quality and efficiency of the processes of information and automation solutions to multidisciplinary problems. This flaw is due to the heterogeneity of the information and its processing algorithms, varying degrees of formalization and knowledge of subject areas, users' training level in the field of information technologies, as well as in the areas of related disciplines.

It should also be noted that the modern problem-oriented systems do not provide the opportunity to involve the domain specialists in software development process to address multidisciplinary tasks not only as sources of knowledge on the subject area, but also as direct developers. This problem is primarily due to generally insufficient level of domain specialists training in the field of information technologies, an extremely small amount of available tools for the software development based on meaningful domain terminology.

Based on the authors' experience in developing automation systems to solve technogenic safety problems, hereby an approach to organizing multidisciplinary information space based on ontology is proposed. This approach is based on several principles:

- information space must be decomposed on the basis of the following grounds: study object structure, object state classes reflecting the dynamics of physical processes occurring in the object, multidisciplinary information aspects reflecting classes of critical states with the viewpoints of disciplines involved in addressing domain tasks;
- the basis of the information space is an ontology decomposed into domain and problem areas ontologies, knowledge representation models, visualization models, as well as application ontology, including models of dialogues, custom interfaces, etc.;
- the ontology must provide presentation of heterogeneous by formalization degree information on the problem addressed;
- ensuring independence of data and knowledge presentation of their processing methods;
- ensuring the team involvement in the development of information space;

- the use of existing software as the basis for creation of individual components of the problem-oriented system developed.

As stated above, ontology is to be the basis of the information space description. This viewpoint is presented in numerous publications.

The paper [1] stipulates, as a result of many studies and applications of ontologies, that ontology occupies a central place in any information system knowledge system, including intellectual one.

The paper [2] states that, in general, to support the decision-making process, ontology performs the following key functions: identifies a common terminology database for all participants of the interacting processes; enables sharing of knowledge represented in the ontology to analyze the new problem situations; provides joint representation of knowledge from various sources through formulation of rules and precedents, using the same domain concepts.

The main advantage of using ontologies in organizational management is a holistic approach to processes controlled. This ensures achievement of the following: consistency (ontology represents a holistic view of the subject area), uniformity (the material presented in a uniform manner is much better perceived), complexity (ontology construction facilitates restoration of missing logical relationships) [3]. The authors also note that sharing ontologies becomes possible as a consequence of joint intellectual space for description of interacting processes control tasks, their solution models and methods, as well as the rules and precedents of decision-making embedded in the ontology.

2. Ontology concept

Description of the concept of the ontology proposed uses notions of domain and problem area ontologies, ontology of presentation and top-level ontology forming ontology hierarchy (Fig. 1), various aspects of which were presented in papers [3-11].

The ontology content is determined on the basis of the principles outlined in the introduction, i.e. information space is classified according to multidisciplinary information aspects (research aspects), object state classes relevant to the tasks (Fig. 2), as well as the disciplines to which the tasks refer (Fig. 3). In turn, the task components, regularities, and methods are also defined by information structures (Fig. 4-5).

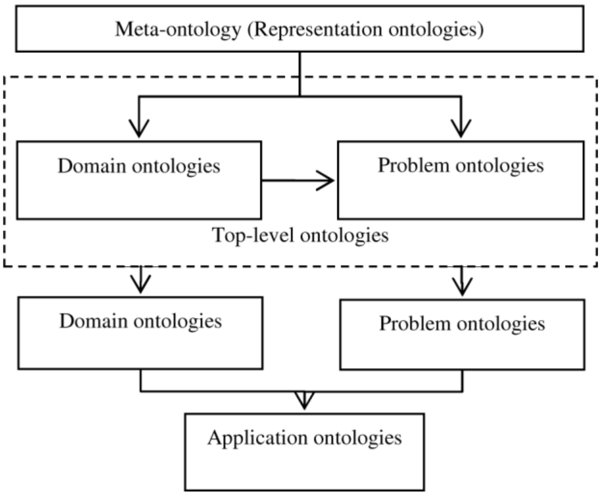

Fig. 1. Ontology hierarchy.

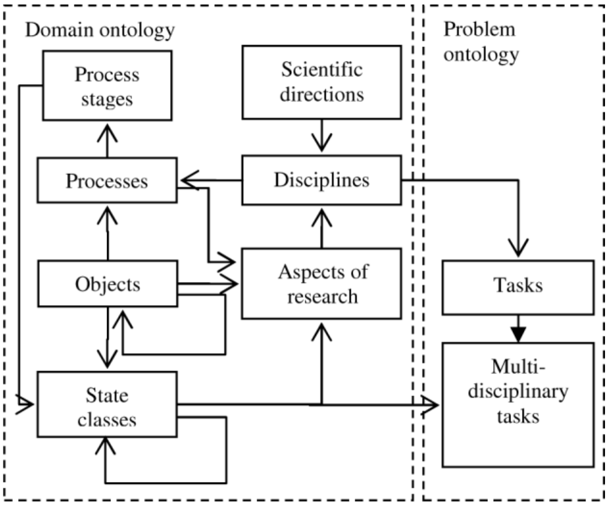

Fig. 2. Information structure describing the object of research.

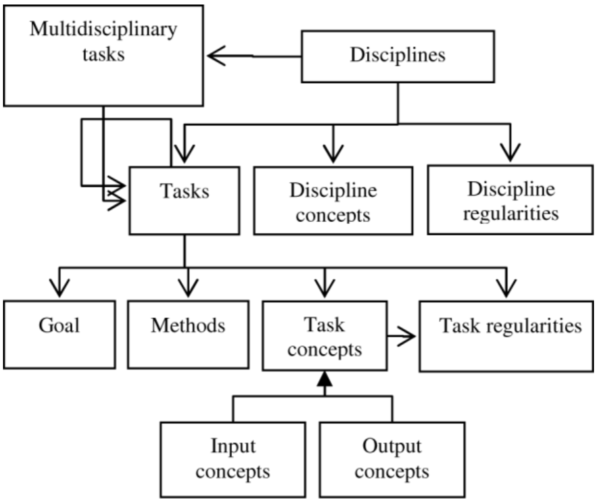

Fig. 3. Information structure describing the discipline.

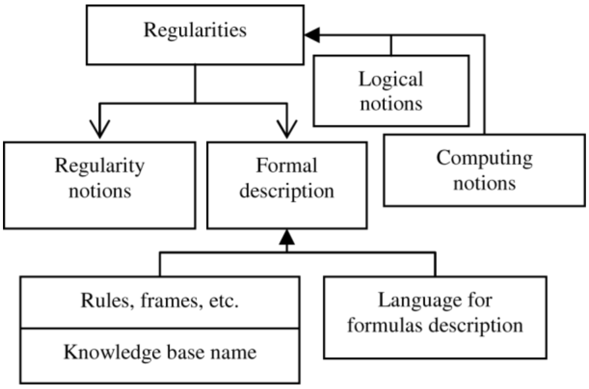

Fig. 4. Structure of information describing regularities.

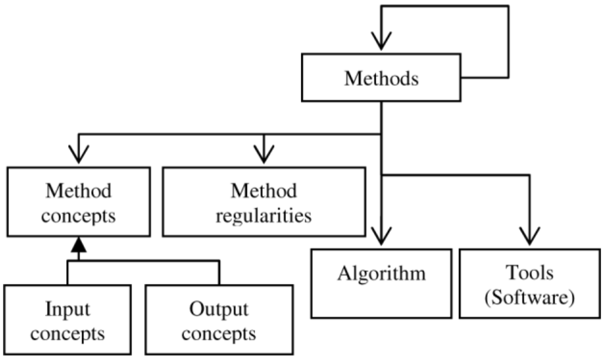

Fig. 5. Structure of information describing methods.

3. Ontology model

The proposed ontology is based on classical model, which contains concepts (notions, classes, entities, categories), concepts properties (slots, attributes, roles) and relationships between concepts (depending of the function). Additional restrictions that are defined by the axioms are not considered hereby.

Ontologies can be arranged in some hierarchy and have different levels of generality, they may be merged (integrated), etc. [6]. The ontology of a higher level of generality may be detailed at a later stage of ontology creation.

We hereby present an ontology as a hierarchy of ontologies, where top-level ontology is highlighted (Fig. 6). This level is intended to describe the fundamental concepts for a variety of domain and problem areas.

To ensure multidisciplinary character of task solution, let us identify research object and discipline ontologies as top-level ontologies (Fig. 1-4). The models of these ontologies are described below.

3.1. Metaontology

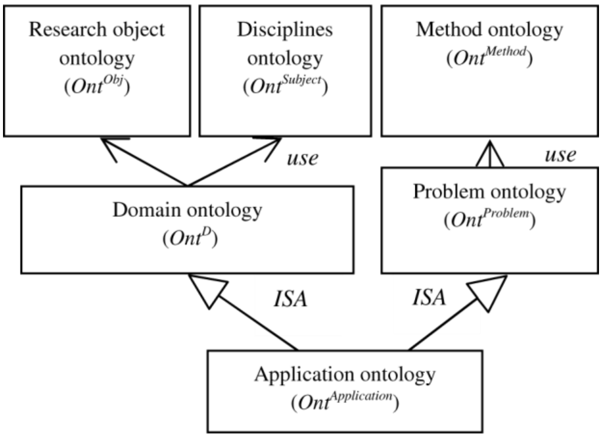
Research object ontology (Ont^{Obj})
Disciplines ontology ($Ont^{Subject}$)
Method ontology (Ont^{Method})
use
use
Domain ontology (Ont^{D})
Problem ontology ($Ont^{Problem}$)
ISA
ISA
Application ontology ($Ont^{Application}$)

Fig. 6. Ontology interaction chart.

Metaontology model includes subject entities (classes and class's specimens) and relations between them [6, 7]:

$$Ont = \langle BT\_D, CN, Pr, Obj, R \rangle,$$

where Ont is metaontology, BT_D - a list of basic data types, $BT\_D =$ {literal, object, collection}, CN - class names, Pr - names of class properties, Obj - concepts (constants, objects), $R = (R_{isa}, R_{pr}, R^{user})$ - ontology relations, in particular, R_{isa} - an inheritance relationship between classes CN, R_{pr} - relationship between classes and properties, $pr \in Pr$, $cn \in CN$, $cn\_bt \in CN \cup BT\_D$, the latter meaning that a property named pr_i characterizes class cn_j, and property values may be represented by elements of another class type from CN or primary multitude of types BT_D, R^{user} - a multitude of relationships between concepts defined by the researcher and later on considered basic. In general, the relationship looks as follows:

$R = \{r_1, ..., r_N : \forall r_i = (rn_i, bt\_r_i, lcn_i, rcn_i, mnr_i)\}$,

where rn_i is a basic type of relationship, lcn_i, rcn_i are the left and right sides of the relationship, referencing to the concepts, mnr_t - interpretation method of the basic type of a relationship. This study identifies the following basic types of relationships: $R = BT\_R = (R_{isa}, R_{pr}, R_{has}, R_{alt})$, R_{has} - relationship «has a part», R_{alt} - relationship of alternatives.

3.2. Domain ontology

Ontology of some subject area includes substantive entities (classes and class specimens) and relations between them for a specific subject area:

$$Ont^D = \left\langle BT\_D, CN, Pr, Obj, R^D \right\rangle \subset Ont \ .$$

3.3. Disciplines ontology

Disciplines ontology is intended to describe (aggregate) information related to the discipline considered, concepts, regularities, tasks:

$$Ont^{Subject} = \left\langle SN, CN^S, P^S, MP^S, L^S, R^S \right\rangle,$$

where SN is the name of the discipline, P^S - discipline tasks that may have basic type sub-tasks, MP^S - multidisciplinary tasks highlighted as a particular kind of tasks using ontologies of various disciplines, CN^S - a multitude of discipline concepts, L^S - description of regularities postulated by the discipline, $L = (CN^L, R^D, LF, LL), LF : (CN^L, R^D) \to LL$, CN^L - a multitude of concepts used to describe regularities, LF - some rules of translation of ontology elements into a formal language LL , $CN^L \subseteq CN$, R^S - a multitude of relations between disciplines, concepts, aims and regularities, in particular $R^S = (R^S_{NS-CN}, R^S_{NS-P}, R^S_{NS-L})$. On the basis of the list of basic relationship types, we can specify the type of relations in the discipline ontology: $R^S \subset R_{has}$.

3.4. Research object ontology

Research object ontology is designed to describe (aggregate) information on the object, the study of which is a multidisciplinary problem. Another peculiarity is the study of the object dynamics based on the study of the processes occurring at the object. Multidisciplinarity ensures involvement of various aspects of the study and the process-oriented approach necessitates description of the processes and their dynamics (Fig. 5):

$$Ont^{ObjR} = \left\langle ObjN, Prc, SPr, CSObj, Asp, SN, R^{ObjR} \right\rangle,$$

where $ObjN$ is the name of the study object, Prc - research object processes, SPr - processes stages, $CSObj$ - object state classes, Asp - research aspects, SN – disciplines, R^{ObjR} - relationship between object, processes, state classes, aspects and disciplines $R^{ObjR} \subset R_{has}$.

3.5. Ontology methods

To describe problem ontology, we shall first describe methods ontology designed to describe various methods without taking into account any information on domain and problem areas. In fact, this ontology describes types of methods that can later be used to create problem ontology and application ontology.

$$Ont^{Method} = \left\langle MN, BT\_M, Input^M, Output^M, L^M, Alg, Soft, Q, R^M \right\rangle$$

where Ont^{Method} is methods ontology, MN method name, BT_M - the basic ontology methods, $Input^M$ - input (source) data and knowledge of the method, $Output^M$ - output data and knowledge gained as a result of the method application, $Input^M \cup Output^M \subseteq CN$, L^M - regularities defined by the method, Alg - method algorithm, $Soft$ - a multitude of software that implements the methods, Q-a multitude of indicators describing the method quality, such as speed, accuracy, expert preferences etc., $R^M = (R^M_{MN-L}, R^M_{MN-CN}, R^M_{MN-Alg}, R^M_{BT\_M-Soft})$ - a multitude of relationships that define the correlation between the following entities: method and regularities, method and concepts, method and algorithm of its implementation, basic method and software of its implementation, respectively, $R^M \subset R_{has}$. Let us specify algorithm description: $Alg = (Item, R^A)$, $Item = (NItem, BT\_Item, BT\_P, BT\_M)$ - algorithm stages, $NItem$ - name of the stage, $BT\_Item = (BTI\_P, BTI\_If, BTI\_L, BTI\_Agr, BTI\_Div)$ - basic type of algorithm stages including the stage of tasks addressed by basic methods, stages of the cycle type, conditions, aggregation and segregation, BT_P is a basic task type, $R^A = (R^A_S, R^A_L, R^A_{Agr}, R^A_{Div}, R^A_{If})$ - a multitude of relations between algorithm stage: sequence, cyclic nature, aggregation, segregation and conditions respectively, relationship data complement a multitude of model relations $R^A \subset BT\_R$.

3.6. Problem ontology

Problem area models embrace knowledge on the ways to address various types of tasks. At the same time, effective solution of each task requires adequate representation of data and knowledge, as well as algorithms that implement the solution. Presented below is the formal description of task ontology:

$$Ont^{Problem} = \left\langle PN, BT\_P, Goal, CN^P, L^P, MN, R^P \right\rangle,$$

$$Ont^{Problem} R^{Ont}_{use} \, Ont^{Subject},$$

$$Ont^{Problem} R^{Ont}_{use} \, Ont^{Method},$$

where $Ont^{Problem}$ is the ontology of problem area, PN - a multitude of tasks; $Goal$ - goal of the task, CN^{P} - task concepts, $Input^{P} \cup Output^{P} \subseteq CN^{P}$, $Input^{P}$ - input (source) data and knowledge of the task, $Output^{P}$ - output data and knowledge gained as a result of the task solution, MN - method names; L^{P} - task regularities, $R^{P} = (R^{P}_{P-CN}, R^{P}_{P-G}, R^{P}_{P-L}, R^{P}_{P-MN}, R^{P}_{P-P})$ - relationship between tasks and concepts, regularities, methods, $R^{P} \subset R_{has}$, R^{Ont}_{use} - use relationship, which in this case means that ontology uses to identify another ontology, i.e. ontology of tasks to determine its components uses concepts from ontology methods and disciplines.

3.7. Application ontology

Ontology application inherits the domain class hierarchy (Fig. 1) described in domain ontology, as well as tasks and solutions created in problem area ontology:

$$Ont^{Application} R^{Ont}_{isa} Ont^{Problem},\ Ont^{Application} R^{Ont}_{isa} Ont^{D}.$$

$$< Class >^{D}\ R_{isa}\ < Class >^{Application}$$

$$< Class >^{Application} = < Class >< Property >< KnowledgeDomain >^{D} + < MethodName >$$

where + - the sign to indicate that the concept preceding it is complemented by the concept placed after it.

In application ontology the description of class properties is complemented by the ability to specify names of computation methods of properties values, and description of tasks and methods is specified, i.e there are given task names and methods for an application, there are defined goals, input and output data on the basis of the domain classes.

4. Ontology example

The approach proposed is tested to address the tasks of complex technical systems security management in the course of development of prototypes of the research automation system agent simulation system. It is hereby exemplified by a fragment of ontology in the form of a list of concepts:

Discipline - material science.

- Task - prediction of submicrolevel defects.
- Goal - determination of acceptability of predicted submicrolevel defects.
- Input concepts - elements of technical system, material and its properties, loads, environment.
- Output concepts - submicrolevel defects and their properties.
- Regularities - regularities, which determine cause-and-effect relationships between material production technology and technical system element, environment and loads, to which technical system element is exposed, and possible submicrolevel defects.
- Methods - an expert system, which identifies possible submicrolevel defects based on the data on environment and loads.
- Discipline - technogenic safety.
- Task - definition of emergency risk.
- Goal - determination of acceptability of emergency risk.
- Input concepts - elements of the technical system, critical failures of technical system elements, failure probability, failure frequency, emergencies, consequences of emergencies.
- Output concepts - emergency risk.
- Regularities - statistical risk assessment, where the magnitude of risk is determined as the sum of products of the probability of a catastrophic (critical) failure of technical system elements by its consequences.
- Methods - computational procedure that determines risk.

5. Conclusion

The paper describes an ontology model, which is instrumental in describing multidisciplinary information on the task considered, including methods and means of its solution. Ontology model is based on a hierarchy of domain ontologies and problem areas. The model ensures the following: presentation of heterogeneous by the degree of formalization information on the problem addressed; independence of data and knowledge presentation of their processing methods; team involvement in the development of information space; the possibility of applying a model-oriented approach to applications development [12]. The model proposed is used to address tasks of complex technical systems safety management within the framework of development of prototypes of research automation system and simulation modeling agent system.

Acknowledgment

The reported study was partially supported by project IV.38.1.2 (AAAA-A17-117032210079-1) and RFBR projects 18-07-01164, 18-08-00560.

References

1. S. Staab and R. Studer (eds), *Handbook on Parts* (Springer, Berlin, Germany, 2009).
2. I. P. Vladimirova, A. I. Malakhova and L. R. Chernyakhovskaya, Analytical support of management decision-making using the ontology of tasks, models and optimization techniques, in *Materials of XII All-Russian Meeting on Management Problems, VSPU-2014* (Moscow, 16-19 June 2014). (Moscow, ISP RAS, 2014), pp. 3929−3935.
3. T.A. Gavrilova, D.V. Kudryavtsev and D.I. Muromtsev, *Knowledge Engineering. Models and methods* (SPb., Publising House «Lan'», 2016).
4. N. Guarino, Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration, in *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, ed. M. T. Pazienza (Springer Verlag, 1997), pp. 139−170.
5. R. B. Studer, R. and D. Fensel, Knowledge Engineering: Principles and Methods, *Data and Knowledge Engineering.* 25 (1-2). (1998) 161−197.
6. A. V. Mantsivoda and A. V. Malykh, *Representation and processing of knowledge in the Internet: Information systems and logic* (Irkutsk, Irkutsk University Publishing House, 2005).
7. Y. Sure, S. Staab and R. Studer, Ontology engineering methodology, in *Handbook on Ontologies*, eds. S. Staab and R. Studer (Springer, Berlin, 2009), pp. 135−152.
8. R. Plant and R. Gamble, Methodologies for the development of knowledge-based systems, 1982–2002, *The Knowledge Engineering Review*. **18** (1) (2003) 47−81.
9. N. Guarino, Formal ontology and information systems in *First International Conference on Formal Ontology (FOIS 1998*), (Trento, Italy, June 1998).
10. R. Kishore, R. Sharman and R. Ramesh, Computational ontologies and information systems: I. Foundations, *Communications of the Association for Information Systems*. 14 (Article 8) (2004) 158−183.
11. O. A. Nikolaychuk, Modeling knowledge to study the technical state dynamics of unique objects, *Control Sciences*. № 4 (2009) 58−65.
12. C. Atkinson, and T. Kuhne, Model-driven development: a metamodeling foundation, *IEEE software.* **20**(5) (2003) 36–41.