Ontological Infrastructures for Intelligent Problem Solving

Valeriya V. Gribova[1,2], Elena A. Shalfeeva[1]

[1] IACP FEB RAS, Radio,5, Vladivostok, 690041, Russia
[2] FEFU, 10 Ajax Bay, Russky Island, Vladivostok, 690922, Russia

Abstract

The main problem solved by the authors is to reduce the complexity of creating and maintaining systems with knowledge bases and increase their viability. The authors see its solution in the classification of intellectual tasks, which allows us to determine the place of the task to be solved among the many known tasks and ready-made solutions. Unified formal problem statements provide the basis for finding ready-made solvers and designing reusable system components. The ontological solver, together with the user interface, is repeatedly used for a variety of systems with knowledge bases, since it does not depend on the content of knowledge bases. So the quality of the created intelligent system can be infinitely improved by integrating more and more complete versions of knowledge bases with the solver.

Keywords

knowledge base; ontological solver, knowledge base systems, infrastructure for building systems

1. Introduction

The development of intelligent systems (IS) has moved from the stage of theoretical research with the implementation of demonstration prototypes to the stage of practical use. During the practical use of the intelligent system, it is expected that it will solve real, rather than simplified, tasks and be used for as long as possible. At the same time, the costs of its development and maintenance will be paid off by the real benefits that its use will bring. However, according to the research company Gartner, only 53% of the prototypes of artificial intelligence systems have a real practical use. The main problem is in the complexity of scaling solutions in the artificial intelligence field, since the available tools are not focused on creating complex systems and managing their life cycle [1].

To reduce the complexity of creating and maintaining software systems (SS), the following mechanisms are known and widely used: architectural solutions (separation into low coupled components with logically understandable functions); declarative representation of SS components; automation tools for building code fragments; use of standard architectural solutions; division of competencies between different types of developers; reuse of ready-made solutions. Well-known SS engineering methods, supported technological solutions and tools play a significant role in this process. However, when a SS is a knowledge-based system, the proposed methods, technologies and tools are practically not applicable. First of all, because the architecture of such system has an additional architectural component – the knowledge base and a new type of developers – domain experts. These factors change the methods, tools, and technology for developing this class of systems.

Despite significant successes in the field of creating systems with knowledge bases, it must be stated that methods and tools for creating and, above all, their subsequent development, have not been given sufficient attention, therefore the proposed toolkit is applicable only for creating demonstration prototypes. Thus, despite modern advances in software and knowledge engineering, the problem of

Russian Advances in Fuzzy Systems and Soft Computing: Selected Contributions to the 10th International Conference «Integrated Models and Soft Computing in Artificial Intelligence» (IMSC-2021), May 17–20, 2021, Kolomna, Russian Federation
EMAIL:
ORCID:

ensuring the viability of intelligent systems with knowledge bases that can be useful for a long time to specialists solving complex and important problems remains relevant.

2. Problem statement

It is considered that the problem of reducing the complexity of creating and maintaining systems and increasing their viability lies in the field of technological and instrumental support for their development. Nevertheless, it is obvious that such problem for systems with knowledge bases gives rise to many scientific problems. We will highlight the main groups of problems that, according to the authors, are aimed at reducing the complexity of creating systems with knowledge bases or increasing their viability.

Architectural solutions that provide separation into low coupled components with logically understandable functions, declarative representation of components. A characteristic feature of systems with knowledge bases (KB) is the explicit separation of KB as a separate component. However, when creating rule-based knowledge bases (bases of productions), a problem-independent knowledge model is used, and procedural part of the knowledge is embedded in the logical inference machine. This approach is acceptable for areas where the number of rules is measured in hundreds. Since the production model is inconvenient and often incomprehensible to specialists, the development of real databases by them becomes an extremely laborious process [2-4], and modification of the databases is practically uncontrollable.

The use of ontologies as a semantic superstructure greatly simplified the process of knowledge formation and provided them with the property of declarativeness. But if we declaratively describe only the properties of entities, their binary relations and identify the class hierarchies of such entities, then an important part of knowledge for decision-making remains outside the semantic representation. An additional problem is the "bringing together" of ontology and knowledge in a single information "block" or resource . This prevents autonomous improvement of knowledge (changing knowledge leads to the need to change software solver).

Inclusion of domain experts in the design process. The use of ontologies describing the domain terminology, the structural connections between concepts, is aimed at including domain specialists in the development process. But in the knowledge model, the role of not only structural relationships is important, but also spatial, causal, temporal and others. The object-oriented model for the presentation of ontological knowledge is not very adapted to the expression of such relations. Moreover, object-paradigm is inconvenient and incomprehensible to most specialists.

A well-known solution is the semantic (graphical) model of knowledge representation. It provides a description of any semantic relationship, not necessarily binary ones. And its subspecies - hierarchical semantic net - additionally facilitates the formation of complete documents and blocks of knowledge consistently "from top to bottom".

Reuse of standard architectural solutions, standardization of services and interfaces, reuse of ready-made solutions (using a library of reusable components). The experience of creating systems with knowledge bases (KBS) using libraries of solved tasks for their reuse has not become effective, since the solutions are related to particular tasks and do not help in automating new tasks and activities. At the same time, in the field of solving intellectual problems, the idea of dividing them into classes and formulating statements has been discussed for several decades. It was assumed that a complete review of such problems could allow us to develop methods for solving them and move in this area from the art of developing knowledge-based systems to technology. At the same time, the presented classifications of intellectual tasks were not based on uniform principles, each author described a certain set of tasks in "his or her" terms, most often in the form of a text, without formal statements.

Taking into account the above, the authors see the achievement of this problem through the solution of the following tasks:

- Classification of intellectual activity tasks solved on the basis of formalized knowledge, description of their formal characteristics, semantic structure of knowledge and its processing algorithms;

- Development of an ontological approach to the formation of declarative knowledge bases, based on the explicit separation of ontology and the knowledge base, their semantic representation.
- Development of methods for declaring and constructing ontological solvers with the reuse of ready-made solutions;
- Formation of a fundamentally important set of infrastructure components for building systems with KB and for their continuous adaptation to the development of the domain.

3. Intellectual task statements

Depending on whether the results of observing the situation or the conditions for solving the problem are set as input data, it is possible to distinguish between the tasks of analyzing the results of observations and the tasks of analyzing the conditions (constraints) for solutions. For those and others, systems with knowledge bases have being created to help specialists. The specializations of pairwise combinations of these abstract problems are of practical importance.

The problem of finding hypotheses that explain the results of observations is considered as a combination of the problem of finding hypotheses and the problem of analyzing the results of observations, in which the KB and the results of observing the situation are given; it is required to find all hypotheses corresponding to the results of observations and formalized knowledge.

The problem of searching for hypotheses explaining the results of observations is considered as a combination of the problem of searching for hypotheses and the problem of analyzing the results of observations, in which KB and the results of observing the situation are given; it is required to find all hypotheses corresponding to the results of observations and formalized knowledge.

The design problem is considered as a combination of the problem of finding hypotheses and the problem of analyzing constraints on the solution. Here, KB and a non-empty set of constraints on the result of solving the problem are given; it is required to find all projects that correspond to the KB and constraints on solving the problem.

The task of criticism of the hypothesis based on the analysis of the results of observations (the KBand the results of observation of the situation and the hypothesis are given and it is necessary to establish whether it corresponds to the knowledge base) and the task of criticism of the project, in which it is necessary to check the compliance of the project with the knowledge base and the restrictions that it must satisfy are similar.

In many domains there are classes of situations and tasks that take into account these classifications: search for hypotheses about classes, criticism of hypotheses, formation of knowledge about classes based on a training set of situations represented by the results of observations. They are considered as a refinement of the above-mentioned abstract problems.

In many domains, complex objects or systems consisting of components are analyzed or designed, and problems are solved that take into account these components and the relationships between them (including spatial and temporal ones). The tasks of analysis, hypothesis criticism, design, project criticism can be set for dynamic or static systems; in the problems of designing a dynamic system, there are statements and formulas that take into account the temporal aspects of a developing or functioning system.

Forecasting and monitoring tasks - refinement of the task of searching for hypotheses that explain the results of observations. In the monitoring task, it is necessary to determine those observations of a dynamic system that can be used to track critical states of the system. In many domains, problems are solved that consider actions (ordered at least partially) that should lead to some given goal.

In a number of domains, problems are solved for systems in which cause-and-effect relationships are related to time. The processes occurring in such a system can be divided into external (observed) and internal.

In accordance with this classification of problems, a hierarchy of their statements is formed, expressed in terms of unified mathematical abstractions for all meaningful concepts.

Examples of formal problem statements.

The major components of *Forecast task* can be summarized as follows.

Givens:

- Kn (knowledge base);

- an ordered set of time moments $\{t_0, \ldots, t_k\}$;
- R $(t_0, ..., t_k)$ – results of observation of signs, characteristics of the system;
- $t' \in \{t_{k+1}, \ldots t_m\}$;
- name of the requested sign q';

Goals:

- Find all such values of q' (..., t',...), for each of which there exists such possible causal models AS (<R($t_0, ..., t_k$), q' (..., t',...)>) that all sentences from Kn are true.

The major components of *Diagnostic task* can be summarized as follows.

Givens:

- Kn (knowledge base);
- an ordered set of time moments $\{t_0, ..., t_k\}$;
- R $(t_0, ..., t_k)$ – results of observation of signs, characteristics of the system and the events that occurred;

Goals:

- Find all possible causal models (AS(R)) of the system, consistent with R, for which all the sentences from Kn are true.

This classification covers almost the entire range of tasks related to the development of knowledge-based systems. At the same time, the classification is open, allowing the introduction of abstractions of new concepts and new tasks into it. In practice, both tasks of one of the classes and compositions of several such tasks need information support.

To identify common, reusable methods, algorithms and procedures for solving tasks from the presented classification, their analysis is carried out using generalized ontologies of relevant knowledge.

For example, the ontology of knowledge about variants of signs of a class of situations, about variants of possible stimuli (reasons) for the appearance of a process, about variants of reaction to events allows us to describe knowledge about a developing process (or a dynamic situation) for solving problems of recognizing a class of a critical situation, determining the stage of development of the process, forecasting the state. The development of the process depends on some signs or characteristics (e.g., length, age). It is noted that we have to deal with processes whose main "parameter" is time, observation of them is, in particular, time series.

In the areas and tasks of analyzing the results of observations, the process developing over time may deviate from the normal case, and the controlled system may deviate from normal functioning. All types of connection of such deviations with causes, factors, external manifestations should be supported by the ontology. In the generalized ontology of cause-and-effect relationships, the main types and subtypes of sentences are defined to represent the relationships of deviations in the functioning of the system (object) with causes, factors, and manifestations.

Such ontology is used to solve diagnostic problems, problems of requesting additional information for recognition, "simple" control problems, forecasting the consequences of control actions. For them, the knowledge model additionally includes proposals of changing the properties, attributes of situations (of each class) if there are deviations from the norm.

Methods and algorithms for solving problems using the same ontology will have the same steps\stages\fragments. Their identification and systematization will allow us to bring the development of ontological solvers to a new level through a wide range of opportunities and ways to reuse existing and systematically accumulated experience (previously proposed and newly created solutions).

A common set of computational operations required for the implementation of problem solvers of different types is revealed: to check the fulfillment of the conditions for considering the hypothesis-diagnosis; to check the fulfillment of the conditions of the sign complex; to find the observation facts for the sign complex; to check the correspondence of the observation results to the variants of the dynamics of the sign, etc.

Algorithms of software components (specialized) solvers for such problems are proposed to be created using specified operations.

4. Ontological model of knowledge

The ontological knowledge base means that the domain ontology includes as part of the ontology of knowledge about the relationships between concepts of different types that are taken into account when solving problems. These are relationships about variants of signs of a class of situations, about variants of possible stimuli (reasons) for the appearance of a process, about variants of reaction to events, etc. Such an ontology is represented as a separate resource, in a semantic (understandable to a person - a knowledge engineer and an expert) representation. A hierarchy of classes of concepts (the Protégé approach [5]) is not enough to solve problems other than classification. But the hierarchy of the structure in the ontology of the description of entities and phenomena is appropriate, because it facilitates the work of the expert, making it possible to gradually move from general concepts to details, elements that clarify general concepts and entities. At the same time, it ensures the integrity of the created documents.

Thus, the main characteristics of ontological knowledge models are [6]:

- explicit separation of the ontology from the knowledge base, ensuring compliance between them; when changing the ontology, all knowledge bases generated on its basis should be automatically brought into a form consistent with the changed model.
- declarative graph (semantic) representation of the ontology and knowledge base, which allows developers to describe the necessary knowledge base "from top to bottom", gradually moving from general concepts to details, elements that clarify general concepts and entities, while the description itself should be holistic, and not "fragmented" into separate variables and entities;
- description of the ontology in a declarative language, which should be quite simple and at the same time powerful, allowing you to describe arbitrary ontologies, oriented and adapted to the terminology and form that is understandable to developers of components of systems with KB;
- automatic generation of editors of knowledge bases on ontology, which provides an opportunity for experts to form and maintain knowledge bases without intermediaries.

Since the domain knowledge is dynamic (they are expanded and adjusted), it is important not only to separate declarative knowledge from the procedures for their use, but also to provide conditions for the development of KB during the entire process of use without the participation of programmers, and preferably without knowledge engineers. In [7], the authors describe in detail the ontological model of knowledge that has these characteristics.

5. Ontological solver

To reason, put forward (generate) and explain hypotheses based on formalized knowledge, the Solver needs to bypass the Knowledge Base (based on knowledge of its structure and ontology as a whole).

The solver processes input information, comparing knowledge to input data (facts or conditions), puts forward hypotheses based on ontological conventions, sentence structure and restrictions on interpretation. At the same time, the Solver builds an explanation, fixing the steps of accepting or rejecting possible logical conclusions.

Its logical inference, in contrast to logical inference implemented as calculus, is created as an algorithm for sequential confirmation or refutation of sub-goals corresponding to the elements of formalized knowledge. This is done by comparing facts to knowledge elements based on ontological conventions and their analysis. The logical inference of the solver for planning, design, and control problems is a sequential check of the sub-goals corresponding to the elements of knowledge, and the search for the elements or actions, the complete set of which, consistent with the ontology, becomes a hypothesis.

The solver, which puts forward and explains hypotheses, is designed from program units of different types, with and without access to information components, for calculations or communication with the external environment.

The program units for searching facts get the values of observations, a list of conditions (events and other factors), search for facts with such names in the input description (document), compare and give the answer.

It is convenient to distinguish between the program units of the "upper layer" - for intermediate conclusions such as confirming or refuting sub - goals when" working out hypotheses "and the program units of the" next layer " -searching for information (among the values of observations or target parameters-restrictions), as well as selecting elements of formalized knowledge (from the described sets or variants).

Since the Solver implements the procedural part of knowledge (or rather, knowledge about the correct use of declarative knowledge), its development (as a software subsystem) corresponds to the traditions of programming technology. The orientation of its components to the domain ontology becomes the main "condition" for the independence of its development and maintenance processes from the processes of creating and developing KB.

Designing the architecture of problem solvers in accordance with the type of problem being solved and the set of structural and causal relationships between entities under consideration, assembling the solver from replaceable software components - ensure the transparency of its architecture. This is especially important in the conditions of collective development of systems with KB and their development.

6. Components of the ontological infrastructure

A typical infrastructure – an environment for the development and development of systems with KB – includes:

- the domain ontology (including the database of terms),
- KB and database editors generated (based on the domain ontology),
- tools for declaring the solver and program units, as well as an environment for coding them,
- a lot of prepared and tested information components (KB, archives and databases);
- a library of software units that process information components;
- a set of problem solvers created from program units.

The extended environment for the development and development of systems with KB includes software for evaluating KB, inductive formation of KB and other software for their improvement.

The technology of KB systems development is implemented using a cloud-based development environment on the IACPaaS cloud platform [7]. On its basis, a number of ontological infrastructures have been developed for solving intellectual problems in various domains – medicine, mathematics, computer security, laser additive manufacturing, etc.

7. Conclusion

The main problem discussed by the authors in the article is reducing the complexity of creating and maintaining systems with knowledge bases and increasing their viability. The authors see its solution in the classification of intellectual activity tasks based on the principle of complicating the properties of domains and determining the place of the solved problem among the many known problems and ready-made solutions. Unified formal problem statements provide a basis for finding ready-made solvers and designing reusable system components. The ontological solver together with the user interface is reused for many systems with knowledge bases. It does not depend on the content of the knowledge bases (it uses it as a parameter), so their completeness and quality can be constantly improved by integrating with the solver into new and different versions of useful systems. This makes it possible to improve knowledge bases without the participation of programmers, i.e. in the conditions of already existing software components.

8. Acknowledgements

This work was supported in part by the Russian Foundation for Basic Research (projects nos. 20-07-670 and 19-07-00244).

9. References

[1] Gartner Identifies the Top Strategic Technology Trends for 2021, 2020. URL: https://www.gartner.com/en/newsroom/press-releases/2020-10-19-gartner-identifies-the-top-strategic-technology-trends-for-2021.

[2] M. Islam, V. Katiyar, Development of a software maintenance cost estimation model: 4th gl perspective, International Journal of Technical Research and Applications 2(6) (2014) 65-68. e-ISSN: 2320-8163. https://pdfs.semanticscholar.org/b117/226e62ea109251ff800b9db286d58a0ed623.pdf

[3] T. J. Lydiard, Overview of current practice and research initiatives for the verification and validation of KBS, The Knowledge Engineering Review 7(2) (1992) 101-113.

[4] M. Wilson, D. Duce, D. Simpson, Life cycles in software and knowledge engineering: a comparative review, The Knowledge Engineering Review 4(3) (1989) 189-204. doi: 10.1017/S0269888900005026.

[5] M. Musen, The Protégé Project: A Look Back and a Look Forward, AI Matters 1(4) (2015) 4-12. doi: 10.1145/2757001.2757003.

[6] V. Gribova, A.Kleschev, Ph. Moskalenko, V. Timchenko, E. Shalfeeva, Extensible Toolkit for Creating Viable Knowledge Base Systems, Rasshiryaemy`j instrumentarij dlya sozdaniya zhiznesposobny`kh sistem s bazami znanij (Software Engineering) 9(8) (2018) 339-348. doi: 10.17587/prin.9.339-348.

[7] V. Gribova, A. Kleschev, Ph. Moskalenko, V. Timchenko, L. Fedorischev, E. Shalfeeva, The IACPaaS Cloud Platform: Features and Perspectives, Second Russia and Pacific Conference on Computer Technology and Applications (RPC) (Vladivostok, Russia, 25-29 sept. 2017). IEEE. (2017) 80-84. doi: 10.1109/RPC.2017.8168076.