Ontological Approach to Software Development for Integrated Expert Systems Created on the Basis of the Problem-Oriented Methodology

Galina Rybina [0000-0002-4077-3660], Anatolii Slinkov [0000-0002-6254-2767], Ilya Sorokin [0000-0002-8734-8661]

National Research Nuclear University "MEPhI", Moscow, Russia
gvrybina@mephi.ru, gvrybina@yandex.ru, slinkov.bpc@mail.ru

Abstract. This work is related to the creation of a new technology for the development of integrated expert systems (IES) based on the further development of the problem-oriented methodology and intelligent software environment of the AT-TECHNOLOGY workbench through the integration of an ontological approach to software design for applied IES and methods of intelligent planning and management of IES development processes. with different architectural typology.
The description of prototyping processes for IES based on the use of the basic components of the model of an intelligent software environment with an emphasis on expanding one of the components of the technological knowledge base by developing an applied ontology of typical IES architectures and implementing interaction with an intelligent scheduler is presented.

Keywords: artificial intelligence, integrated expert systems, problem-oriented methodology, AT-TECHNOLOGY workbench, intelligent software environment, automated planning, integrated expert systems' typical architecture, ontology model, applied ontology

1. Introduction

Methods of intelligent planning, and their integration with knowledge engineering, proposed and described in detail in [1-3] and other works, underlie a new technology for building one of the most common classes of intelligent systems - integrated expert systems (IES) [1,2], the demand for which in modern conditions of extraordinary attention to the application of methods of artificial intelligence (AI) technologies has increased significantly, especially in light of the priority areas for the development and use of technologies defined by the Decree of the President of the Russian Federation (No. 490 of 10.10.2019).

Proceedings of the of the XXIII International Conference "Enterprise Engineering and Knowledge Management" (EEKM 2020), Moscow, Russia, December 8-9, 2020.

To build an IES, a problem-oriented methodology has been created, actively used and is constantly developing [1], the essence of which is conceptual modeling of the IES architecture at all levels of considering integration processes in IES and focusing on modeling specific types of non-formalized tasks (NF-tasks), relevant technologies knowledge-based systems (KBS (ES)). The Workbench type toolkit (AT-TECHNOLOGY workbench [1-3]) provides intellectual support for the development of IES at all stages of the life cycle (LC) of building and maintaining an IES (a detailed description of the intelligent software environment of the AT-TECHNOLOGY workbench can be found in [1-3] and in other works).

A significant place within the problem-oriented methodology is given to methods and means of intellectual support for the most labor-intensive stages of the life cycle of building an IES - analysis of system requirements and design. Here, the concept of an intelligent environment model [1,2], the main components of which are a technological knowledge base (KB) and an intelligent scheduler, is used as the conceptual basis of intelligent technology.

The accumulation of experience associated with the development of various IES architectures for specific problem areas and classes of problems to be solved has shown that the processes of prototyping IES have the greatest complexity due to the high laboriousness of building an IES architecture model performed by a knowledge engineer, as well as the need to use a large number of individual software and information components of the AT-TECHNOLOGY workbench, which implement the basic functionality of the IES of various architectural typologies (static, dynamic, training, etc.).

It is also important to note that the complication of management processes for software development of applied IES at certain stages of the life cycle depends significantly on the sources of knowledge used and the integration of various technologies for the automated acquisition (identification) of knowledge from experts, natural language texts (Text Mining technologies), databases (technologies Data Mining, Deep Data Mining, etc.). Since, as a rule, these technologies arose and developed independently of each other, such autonomy and distribution significantly complicates both sharing, leading to an increase in labor intensity in the development, maintenance and monitoring of such voluminous resources as knowledge bases (KB), DB, etc. software tools, including their repeated component-wise use [4].

Therefore, the problems of integrating the methods and technologies of Text Mining, Data Mining, etc., as well as research in the field of creating tools and technologies for distributed knowledge acquisition are most relevant today, as evidenced by a number of works, for example, [5-7] and others.

A similar problem of integrating models, methods and software arises in the design of software for dynamic IES [8], in particular, when using simulation modeling to create a subsystem for modeling the external world (environment) that determines the processes of functioning in real time, as well as in training IES and web IES [9], for the construction of which a significant number of individual software and information components of the AT-TECHNOLOGY workbench [1,2] and others are used.

One of the new approaches to the creation of intelligent technologies for the development of IES is the further development of the problem-oriented methodology and the intelligent software environment of the AT-TECHNOLOGY workbench based on the expansion of the technological knowledge base [1,2] by including in its composition, in addition to the basic components (standard design procedures , reusable components, and other ontologies of typical architectures of applied IES).

Proceeding from this, the main emphasis in this work is made on the discussion of issues related to the experimental software research of prototyping processes for IES based on the integration of the ontological approach and methods of intelligent planning and management [3].

It should be noted that in modern research considerable attention has begun to be paid to both the creation of ontological models of software design processes in general, for example, [10] and others, and the development of methods and tools for ontological modeling of specific design processes, as well as the specification of the systems being developed.

Currently, a new direction is emerging, called Ontology-Based-Software-Engineering (OBSE) or Ontology-Driven-Software-E Engineering (ODSE) [11-17], within which various semantic models for the design of software systems are created, including ontological modeling of software development management processes is carried out.

As a rule, all semantic models of the processes of creating software systems and / or their components are based on a specific classification of the used ontology models, including specific methods for implementing ontological models of design processes, for example [18], etc. In addition, work is actively underway to create instrumental means of ontological engineering, in particular [18-20], etc. Accordingly, in the context of OBSE-ODSE [11-17], of course, it is possible to consider works on the intellectualization of prototyping processes for applied IES and their individual subsystems and components based on the development and the use of an applied ontology of typical IES architectures and tools that ensure its construction and effective interaction with an intelligent scheduler and other basic components of an intelligent software environment model [1-3], which is an integral part of a problem-oriented methodology.

2. Some features of intellectualization of IES prototyping processes

Let us briefly consider some of the important features of the technology for supporting the prototyping processes of applied IES using the main components of the intelligent software environment [1-3].

The basic declarative component of the model of an intelligent software environment in accordance with [1] is a technological knowledge base, containing knowledge about the accumulated experience of building IES in the form of a set of standard design procedures (SDP) and reusable components (RUC). An important operational component of the model by an intelligent program of the environment is the means of intelligent planning of actions of engineers based on knowledge from knowledge, which ensure the generation and execution of plans for constructing prototypes of IES, i.e. an intelligent planner developed on the basis of integrating models and methods of intelligent planning with knowledge engineering methods used in the field of IES [3].

The initial data for generating IES prototype development plans are the IES prototype architecture model, described using the hierarchy of extended data flow diagrams (EDFD [1]), and the technological knowledge base containing a set of SDP and RUC. Accordingly, the IES prototyping process model [3] includes the function of planning the actions of knowledge engineers to obtain the current IES prototype for a specific problem area. The main task of the intelligent planner is to automatically generate plans (global and detailed [3]) based on the IES architecture model and a set of SDPs from the technological knowledge base, which significantly reduces the risks of erroneous actions of knowledge engineers.

The implementation of the tasks of the plan is carried out using a set of operational (instrumental) RUC. Operating the SDP as the main algorithmic element, the intelligent planner at each moment of time makes a detailed construction of the IES development plan, depending on the current state of the project (the type of the NF problem being solved [1], reflected on the architecture model), the features of problem areas, the presence on the architecture model designed drives, etc.

The general architecture of the AT-TECHNOLOGY workbench is built in such a way that all functionality is distributed, i.e. “Spreads” to components registered in the workbench environment and operating under the control of an intelligent development support environment. At present, within the framework of the intelligent technology for constructing an IES, two groups of RUCs are used - components that implement the capabilities of the procedural RUC, and components that implement the capabilities of the information RUC [1,2].

All SDPs are classified as follows [1]: SDPs that do not depend on the type of task, for example, related to the processes of acquiring knowledge from various sources (experts, NL-texts, databases); SDP, depending on the type of task, for example, building the components of training IES; SDPs associated with RUC, i.e. procedures containing information about the life cycle of the RUC from the beginning of its adjustment to the inclusion in the prototype of the IES, as well as information about the tasks solved by this RIC, the necessary settings and, possibly, their values.

The increasing complexity of IES architectures, and the appearance of a large number of RDPs and RUCs in the technological knowledge base led to an increase in the complexity of the search, and therefore the methods and algorithms for planning [3] used by the intelligent scheduler were improved by implementing a fairly well-known approach related state space [21], etc.

The essence of the method for generating an action plan for a knowledge engineer is to perform a sequence of transformations of the architecture model of the current IES prototype, performed in 4 stages [3]: obtaining a generalized EDFD in the form of a graph; generating an exact coverage (i.e., a set of RDP instances with mutually disjoint fragments containing all the graph vertices) using heuristic search; generating a knowledge engineer plan based on the resulting detailed coverage; generating a plan view.

All of the above steps are performed by an intelligent scheduler that fully implements the functionality associated with planning IES prototyping processes. With the help of the preprocessor of the EDFD hierarchy, the preprocessing of the EDFD hierarchy is performed by transforming it into one generalized diagram of maximum detail. The task of covering the detailed EDFD with the existing RDP is implemented using the global plan generator, which, based on the technological knowledge base and the constructed generalized EDFD, ensures the fulfillment of the task, as a result of which an accurate coverage is built, which is subsequently converted into a global development plan.

The detailed plan generator provides detailing of each element of the coverage, i.e. on the basis of the obtained EDFD coverage and technological knowledge base, each element of the coating is detailed, thereby forming a preliminary detailed plan.

Then, based on the analysis of available RUCs and their versions (data on which are requested from the development process control component), the plan interpretation component forms a detailed plan, where each task is associated with a specific RUC and can be performed by a knowledge engineer. With the help of the component for constructing the final plan, the necessary representation of the plan is formed for its use by other components of the intelligent software environment (component of the plan visualization, etc.).

As noted above, the complication of the architectures of the designed IES and the appearance of a large number of RUCs in the technological knowledge base led to an increase in the complexity of the search and increased the negative effect of the non-optimal choice of solutions. Therefore, the composition of the technological knowledge base was expanded by building an applied ontology of typical IES architectures [2].

A detailed description of the methods and algorithms for the implementation of intelligent planning tools is given in [2,3] and others, and below, in the context of this work, some results of experimental modeling based on the implementation of the ontology of typical IES architectures are considered.

3. General characteristics of the basic and modified ontology models considered in the context of the problem-oriented methodology

As a basic model of applied ontology, we took a model developed within the framework of the problem-oriented methodology for constructing IES [1], and quite effectively used to create ontologies of courses / disciplines within the framework of the Chamber of Commerce and Industry “construction tutoring IES”. It is a semantic network described as: $M = <V, U, C>$, where V is a set of elements of ontology elements; $U = \{uj\} = \{<Vkj, Vlj, Rj>\}$, j = 1,…, m is the set of links between ontology elements, where *Vkj* is the parent node, *Vlj* is the child node, *Rj* is the link type, and $R = \{Rz\}$, where z = 1, ..., Z, *R1* is a connection of the "part-whole" type (aggregation), which means that the child node is part of the parent node; *R2* is a link of the "association" type, which means that in order to master the concept of a parent node, it is necessary to own the concept of a child node; *R3* - "weak" connection, means that to own the concept of a parent node, possession of the concept of a child node is desirable, but not necessary; *C* = {*Ci*}, i = 1, ..., a - the set of hierarchical links between the elements of the ontology, while *Ci* = <*Vk, Vl*>, where *Vk* is the parent element, Vl is the child element in the hierarchical structure of the ontology;

The current version of the applied ontology [2, 9] of typical IES architectures is presented in the form Oarch = <Mom, Farch>, where Mom is a modified model of typical IES architectures; Farch is a set of basic and modified operations (procedures) for constructing ontology elements, implemented in the form of software components, each of which, in accordance with the requirements of the intelligent software environment of the AT-TECHNOLOGIES workbench, is designed as an operational RUC.

The modified model is a semantic network described in the form: Mom = <Vom, Uom, PDom>, where Vom is a set of elements of an architecture model (Mies), built on the basis of the ideas of deep integration of components (at all levels of integration), and each element includes name of the ontology vertex, weight (in the range 0 ... 100) and information about the RUCs used; Uom - a set of links of several types between elements of the Mies model (parent and child nodes of the ontology), and the semantics of the types of these links can vary widely (aggregation, association, hierarchy, strong, medium and weak links, etc.) depending on the RDP used; PDom (optional) a lot of special data, i.e. information of a different nature, specifying features and / or non-standard approaches to the development of individual components of the IES prototype (parameters, texts, information about external subsystems, components, applications, etc.).

To build an applied ontology on the basis of this model, tools have been developed to support the construction of an applied ontology of typical IES architectures.

4. Features of the construction of typical IES architectures' applied ontology

As already described earlier, the essence of the problem is to implement an approach to the construction of applied ontologies based on taking into account the features of the architecture model of the designed IES and the features of the currently implemented component-wise functionality in the form of a set of RUCs. The architecture model of the IES prototype is presented in the form of the EDFD hierarchy [1], which is one of the most important components of the project, since its structure largely determines the composition of the prototype and its functionality. The elements of the EDFD hierarchy are characterized by such data types as: NF-operation (NF); Formalized Operation (Op); Essence (E); Storage (S), etc.

The peculiarity of the IES architecture model lies in the multi-level integration, manifested in the EDFD hierarchy and, as a consequence, the identification of architecture elements at different nesting levels, which leads to different architectural solutions, including the use of RUC. Accordingly, the same elements of the architecture can have different functionality; therefore, RUCs that implement this functionality for identical elements also work in different ways.

Therefore, the structure of the applied ontology of typical IES architectures, as well as the algorithms and procedures for its creation and storage, should be developed so that, through their use, it is possible to configure the RUC depending on the structure of the IES architecture model.

On the other hand, the advantage of the basic ontology model lies in the fact that instead of the general vocabulary of concepts used in most information processing systems of the ontological type, a semantic network is used, which made it possible to significantly strengthen the semantics of vertices and display a significant number of relations not only of the taxonomic type, as in most ontologies, but also relations reflecting any declared specificity, as well as using powerful functionality in the interpretation of relations and nodes (elements of the ontology [2,9]. This principle is also used to form the structure of the applied ontology of typical IES architectures, namely: at the top level of the ontology there are various IES architectures (the level of typical architectures); then the architecture elements are located (this level, as follows from the ontology model, contains an unlimited number of sublevels, depending on the nesting of the components that make up the architecture elements); then the operations that are performed by the nty elements of architecture; at the lower level of the ontology, there are RUCs that implement the operations of the components of the architecture elements.

There are three types of links between the elements of the ontology: a link of the "part-whole" type (aggregation), for linking the elements of the ontology that are at different but adjacent levels; a link of the "association" type, for linking ontology elements at the same level; "Weak" link, for linking elements that are both at adjacent levels and at the same level.

In addition to the previously described types of links in the ontology of typical IES architectures, interlevel links between ontology elements located at different levels will be implemented.

Now let's look briefly at some of the implementation features. Since the ontology of typical architectures is part of the technological knowledge base, it has access to data on the RDP and RUC. The technological knowledge base is stored in the form of an XML document [22], in which the methods of describing all elements using tags are determined. First, a description of 4 types of RDP elements is presented: operation - function, unformalized operation - nf function, entity - entity, storage - store. The following is a description of the links between the elements - flow, after which a description of the RDP fragments is given - fragment. At the end, the chronology of the execution of the RDP fragments is described in the case of covering them with the maximum granularity of the EDFD elements as a result of the work of the intelligent scheduler - network. Thus, RUCs must implement the functionality of the RDP fragments in the order in which these fragments cover the elements of the EDFD of maximum detail.

Since the ontology of typical architectures contains various models of IES architectures and their elements (subsystems / tools / components), the following operations can be performed on elements of architecture models, as on elements of an ontology: initialization of adding a new architecture to the ontology; adding architectural elements to the ontology; removal of architectural elements from the ontology; sampling of ontology elements; unification of ontologies.

5. Conclusion

Thus, the experimental base in the form of accumulated information and software for applied IES, developed on the basis of the problem-oriented methodology and the AT-TECHNOLOGY workbench supporting it, turned out to be a successful "testing ground" for the continuation and development of research in the development of elements of a new intelligent planning and control technology. processes of building intelligent systems, including those based on the ontological approach.

In fact, a technological transition was made from "automation" to "intellectualization" of labor-intensive design processes and maintenance of information and software for applied IES, by creating conditions for the effective use of an intelligent scheduler, in particular, to create a technological knowledge base (RDP, RUC information and operational character, ontology of typical architectures), and then carry out full-featured research to create elements of a new technology.

6. Acknowledgments

This work was supported by the Russian Foundation for Basic Research (project no. 18-01-0457).

References

1. Rybina G.V. *Teoriya i tekhnologiya postroeniya integrirovannyh ekspertnyh sistem*. M.: Nauchtekhlitizdat (2008).
2. Rybina G.V. *Intellektual'nye sistemy: ot A do YA. Seriya monografij v 3 knigah. Kniga 1. Sistemy osnovannye na znaniyah. Integrirovannye ekspertnye sistemy*. M.:Nauchtekhlitizdat (2014).
3. Rybina G.V., Blokhin Y.M. Methods and Software Implementation of Intelligent Planning for Integrated Expert System Design. *Scientific and Technical Information Processing*. 46(6):1-12 (2019).
4. Rybina G.V., Slinkov A.A., Buyanov D.R. (2020) The Combined Method of Automated Knowledge Acquisition from Various Sources: The Features of Development and Experimental Research of the Temporal Version. In: Kuznetsov S.O., Panov A.I., Yakovlev K.S. (eds) *Artificial Intelligence. RCAI 2020. Lecture Notes in Computer Science*, vol 12412. Springer, Cham. doi: https://doi.org/10.1007/978-3-030-59535-7_2
5. Aggarwal C. C., Zhai C. Mining Text Data. Springer (2012).
6. Aljawarneh S., Anguera A., Atwood J.W., Lara J.A., Lizcano D.. Particularities of data mining in medicine: lessons learned from patient medical time series data analysis. Eurasip *Journal on Wireless Communications and Networking*, Issue 1. Article 260 (2019).

7. Efimenko I.V., Khoroshevsky V. F. Identification of Promising High-Tech Solutions in Big Text Data with Semantic Technologies: Energy, Pharma, and Many Others (Chapter 16. Advanced Methods). In: Innovation Discovery. Network Analysis of Research and Invention Activity for Technology Management. Ed. by: Tugrul Daim, Alan Pilkington (2018).
8. Rybina G.V. *Dinamicheskie integrirovannye ekspertnye sistemy: tekhnologiya avtomatizirovannogo polucheniya, predstavleniya i obrabotki temporal'nyh znanij*. *Informacionnye izmeritel'nye i upravlyayushchie sistemy*. T.16, №7, S.20-31 (2018).
9. Rybina G.V. *Intellektual'naya tekhnologiya postroeniya obuchayushchih integrirovannyh ekspertnyh sistem: novye vozmozhnosti. Otkrytoe obrazovanie*. 21(4):43-57 (2017).
10. Horoshevskij V.F. *Proektirovanie sistem programmnogo obespecheniya pod upravleniem ontologij: modeli, metodi realizacii*. *Ontologiya proektirovaniya*. 9(4):429-448 (2019).
11. Mendes O. et al. Software engineering ontology: A development methodology. *Metrics News*. 9:68-76 (2005).
12. Happel H. et al. Applications of ontologies in software engineering. In: *Proc. of the 2nd International Workshop on Semantic Web Enabled Software Engineering, ESE' 06*. pp. 1-14 (2006).
13. Bossche M.V. et al. Ontology Driven Software Engineering for Real Life Applications. In: *Proc. 3rd International Workshop on Semantic Web Enabled Software Engineering, SWESE 2007*, Innsbruck, Austria (2007).
14. Nianfang X. et al. Software components description based on ontology. In: *Proc. of the 2nd International Conference on Computer Modeling and Simulation* (Jan. 22-24, 2010). IEEE XPlore Press, pp. 423-426 (2010).
15. Liao L. et al. Software Process Ontology and Its Application. Studies on the Semantic Web. Vol. 17: Semantic Web Enabled Software Engineering. IOS Press; 2014. pp. 207-217 (2014).
16. Pileggi S.F. et al. Ontology in Software Engineering. In: *Proc. Australasian Conference on Information Systems*. Sydney, Australia (2018).
17. Jabar M.A. et al. General Domain Ontology in Enterprise Software Development Process. *International Journal of Engineering and Advanced Technology*. 8(3S):398-402 (2019).
18. Happel H.J. et al. KOntoR: An Ontology-enabled Approach to Software Reuse. In: *Proc. SEKE 2006: the 18th International Conference on Software Engineering & Knowledge Engineering* (July 5-7, 2006, California, USA, 2006). pp. 349-354 (2006).
19. Würsch M. et al. SEON: A pyramid of ontologies for software evolution and its applications. *Computing*. 94(11):857-885 (2012).
20. Olszewska J.I. et al. ODYSSEY: Software Development Life Cycle Ontology. In: *Proc. International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (Seville, Spain 18 Sep 2018-20 Sep 2018). pp. 303-311 (2018).
21. Nau D.S. Current trends in automated planning. *AI Magazine*. 28(4):43-58 (2007).
22. Rybina G.V., Sorokin I.A., Korsachev A.A., Dvorchik M.A. Some aspects of intelligent technology of construction of the tutoring integrated expert systems. *Open Semantic Technologies for Intelligent Systems*. (4):69-72 (2020).