



Some Approaches to Implementation of Intelligent Planning and Control of the Prototyping of Integrated Expert Systems

Galina V. Rybina^(✉), Yury M. Blokhin, and Levon S. Tarakchyan

National Research Nuclear University MEPHI
(Moscow Engineering Physics Institute),
Kashiskoe sh. 31, Moscow 115409, Russian Federation
galina@ailab.mephi.ru

Abstract. The work discusses results of development of basic components of the intelligent software environment of the AT-TECHNOLOGY workbench intended for automation and rendering intelligent the processes of building integrated expert systems (IES) based on a problem-oriented methodology. The emphasis is on aspects of the implementation of the intelligent planner in the context of the problem of intelligent planning of prototyping processes of IES and the proposed method for its solution.

Keywords: Integrated expert system · Problem-oriented methodology
Intelligent program environment · Automated planning
Standard design procedure · Reusable component
Model of processes of prototyping

1 Introduction

Currently, integrated expert system (IES), with a single large-scale development strategies and solutions of various formalized and informalized tasks are most popular in many industrial and socially important areas. An analysis of the experience of developing internationally recognized IES (e.g., [4, 5, 13]), as well as domestic IES, created on the basis of a task-oriented methodology for constructing an IES and supporting the toolkit AT-TECHNOLOGY [7, 8] has shown that the complexity and complexity of the stages of the analysis of requirements and general and detailed design of the IES, and the specific influence of the specific domain and the human factor play a significant role.

The main problems that show the need to create highly effective tools for automation and rendering intelligent the processes of IES development are as follows:

- The impossibility of knowledge engineers (knowledge analysts) to fully determine the requirements for the systems being developed, the lack of reliable

methods for assessing the quality of verification and validation of prototypes of the IES, and the inapplicability of traditional tracing technology to the knowledge base (KB) of the IES;

- A significant increase in the number of intermediate stages and iterations in the life cycle models of the prototype construction of the IES, depending on the architecture of the IES being designed, the types of knowledge sources and the percentage of unreliable and temporal information;
- Practical absence of tools providing not only the automated design of software and information support for applied IES at all stages of the life cycle, but also reducing the intellectual burden on knowledge engineers.

Typically, the vast majority of commercial tools designed for developing intelligent systems, for example G2 (Gensym Corp.), RTWorks (Talarian Corp.), SHINE (NASA/JPL), RTXPS (Environmental Software & Services GmbH), etc. do not “know” what the knowledge engineer designs and develops with their help, so the effectiveness of the application of these tools is completely determined by the art of the developers. In the context of solving the above-mentioned problems, certain results were obtained by approaches such as KBSA (Knowledge Based Systems Assistant) and KBSE (Knowledge Base Software Engineering) integrating, at best, the capabilities of expert systems development tools and CASE tools.

One of the most well-known tool of the new generation, in which the KBSE approach is implemented, which includes elements of “intellectualization” of the processes of developing intelligent systems, is the AT-TECHNOLOGY [7,8] workbench, which not only provides automated support for all phases of the life-cycle building on the basis of a problem-oriented methodology, but also allows us to carry out intelligent assisting (planning) the actions of knowledge engineers through the use of so-called technological knowledge about the standard design procedures (SDP) and reusable components (RUC), which are in aggregate the basic components of the model of the intelligent software environment, the concept of which is described in detail in [7] and other works.

New results in the development of the intelligent software environment of the AT-TECHNOLOGY workbench were obtained by expanding the functionality and increasing the role in the prototyping processes of the IES of the main operational component, an intelligent planner. The analysis of the current versions of the intelligent planner [7,8] showed that with the complication of the models of IES architectures, as well as the appearance of a large number of SDP and RUC in the technological KB, the time and laboriousness of the search for solutions has increased significantly and the negative effect of the non-optimal choice of solutions became more significant.

Therefore, there was a need to improve the methods and planning algorithms used by the intelligent planner. The results of the system analysis of modern methods of intelligent planning and the cycle of experimental studies [9,10] have shown the effectiveness of applying a fairly well-known approach to the implementation of the new version of the intelligent planner related to state space planning.

A detailed description of the basic components of the intelligent software environment of the AT-TECHNOLOGY workbench, intended for automation and intellectualization of the processes of building IES of various architectural typologies on the basis of a problem-oriented methodology, is given in [10, 12], and the focus of this work is a discussion of the general formulation of the task of generating plans for the development of prototypes of IES, approaches to its solution and the features of the software implementation of an intelligent planner.

2 Model of Prototyping Processes of Applied IES

An important feature of the problem-oriented methodology and intelligent software environment of the AT-TECHNOLOGY workbench is rendering intelligent rather complex and time-consuming processes of prototyping in applied IES at all phases of the life cycle, starting from requirement analysis up to creating a series of IES prototypes. To reduce the intellectual burden on knowledge engineers, to minimize possible erroneous actions and time risks in the prototyping of IES, according to [7], it is envisaged to use a technological KB containing a significant number of SDP and RUC reflecting the expertise of knowledge engineers in the development of applied IES (static, dynamic, tutoring).

Accordingly, the formal statement of the problem of intelligent planning of the prototyping processes of IES is considered in the context of the model of prototyping processes of IES in the following form [11]: $M_{proto} = \langle T, S, Pr, Val, A_{IES}, PlanTask_{IES} \rangle$, where T is the set of problem domains for which applied IES are created; S is the set of prototyping strategies; Pr is the set of created prototypes of IES based on a problem-oriented methodology; Val is the function of expert validation of the prototype of the IES, determining the need and/or the possibility of creating subsequent IES prototypes for a particular problem domain; A_{IES} is the set of all possible actions of knowledge engineers in the prototyping process; $PlanTask_{IES}$ is the function of planning knowledge engineer's actions to obtain the current prototype of IES for a particular problem domain. A detailed description of all the components of M_{proto} is given in [11], as well as the specification of some basic components of the model of the intelligent software environment.

For the effective implementation of the $PlanTask_{IES}$ component of the M_{proto} model, an analysis was made of modern methods of intelligent state space planning applied to the investigation of the mechanisms of actions of knowledge engineers in the construction of architecture models (M_{IES}) of different IES [7] at the initial stages of the lifecycle (requirements analysis, general and detailed design). Experiments have shown that the best results are achieved if the search space is formed by modeling the actions of the knowledge engineer while constructing fragments of the M_{IES} model using the appropriate SDPs (for a formal description of this process, graph theory is used by reducing the problem to the problem of covering of the M_{IES} model, represented as a labeled graph, with SDP fragments).

In general, the concept of the M_{IES} model plays a key role in the generation of plans for prototyping IES. In [7], the methods of constructing M_{IES} based on the mapping of interactions of real systems by means of structural analysis expanded by the inclusion of the special element “non-formalized operation” (NF operation), which points to the need of involving experts and/or other sources of knowledge. Therefore, M_{IES} model of an IES prototype is constructed as a hierarchy of extended data flow diagrams (EDFD).

An important role in the implementation of the PlanTaskIES component is played by the plan for constructing a specific coverage (i.e., the sequence of applied SDP fragments), which can be uniquely transformed into a plan for constructing a prototype of IES. In this case, the plan for constructing the prototype of the IES can be represented like [11]: $Plan = \langle A_G, A_{atom}, R_{prec}, R_{detail}, PR \rangle$ where A_G is the set of global tasks decomposable into subtasks; A_{atom} is the set of planned (atomic) tasks, the execution of which is necessary for the development of the IES prototype; R_{prec} is the function that determines the predecessor relation between the planned tasks; R_{detail} is the relation showing the affiliation of the planned task to the global tasks; PR is the representation of the plan.

Thus, with the help of the relation R_{prec} and the sets A_G and A_{atom} two task networks are formed: enlarged and detailed. The enlarged network of tasks obtained with R_{prec} and A_G is called the global plan (the relation of the precedence between the elements of the A_G is obtained on the basis of the R_{detail} detail relation), and the detailed network of tasks obtained on the basis of R_{prec} and A_{atom} is called the detailed plan, with each planned task associated with specific function by a function of a specific operational RUC.

Based on the experience of developing applied IES, a certain limitation is imposed on the structure of SDP: at least one compulsory fragment and an arbitrary number of optional fragments associated with the mandatory fragment of at least one data flow. As a rule, the first element of the set of fragments includes an non-formalized operation (NF-operation), which plays the main role in the construction of the model M_{IES} of the current prototype of IES.

3 Formulation of the Problem of Generation of Plans for the Development of Prototypes of IES

Let us consider the general formulation of the problem of generation of plans for developing prototypes of IES where the initial data is: the model of the architecture of the prototype of the IES (M_{IES}), described using the hierarchy of EDFD; the technological KB, containing a set of SDP and RUC. In addition, a number of restrictions and working definitions is introduced [11]:

1. From the composition of the model M_{IES} only a set of elements and a set of data flows represented in the form of a marked oriented graph G_{RIL} , where the labels determine the relationship between the elements of the hierarchy of the EDFD and the vertices and arcs of the graph. This graph is called a generalized EDFD, and it can be uniquely obtained from the initial model of the architecture M_{IES} .

2. The fragment of the generalized EDFD is an arbitrary connected subgraph contained in G_{RIL} . An SDP instance is an aggregate of TPP and a fragment of a generalized EDFD satisfying the conditions of applicability (the C component of the TPP model) of the corresponding SDP. The cover ($Cover$) of a generalized RDPA is the set of instances of SDP with mutually disjoint fragments containing all the vertices of G_{RIL} (or cover the entire G_{RIL}).
3. A coarse coverage of a generalized EDFD is a EDFD coverage in which all instances of the SDP contain only mandatory fragments. The fine coverage is an extension of the coarse coverage by including optional fragments in the coverage.
4. The inclusion of each fragment of the SDP in the cover is compared with a certain value determined by an expert evaluation.

Thus, the problem of generating a plan for developing a prototype of IES, taking into account the initial data and imposed constraints, is conveniently presented in terms of states and transitions [1–3,6] in the form of a model [11]: $PlanTask_{IES} = \langle S_{IES}, A_{IES}, \gamma, Cost, s_0, G_{IES}, F_{COVER} \rangle$, where S_{IES} is the set of states of the graph G_{RIL} that describe the current coverage $Cover$; A_{IES} is the set of possible actions over G_{RIL} , which are the addition of fragments of specific instances of SDP to the cover (the total set is formed in the aggregate WKB and G_{RIL}); γ is the transition function between states; $Cost$ is a function that determines the cost of a sequence of transitions; s_0 is the initial state describing the empty coverage; G_{IES} is the function of determining whether the state belongs to the target state; F_{COVER} is the function of generating a development plan ($Plan$) from the coverage ($Cover$). The solution of $PlanTask_{IES}$ is the plan of actions of the knowledge engineer (model $Plan$).

To solve the problem, a method was developed (described in details in [11]) with four stages: obtaining a generalized EDFD (G_{RIL}) from the model M_{IES} ; generating an exact $Cover$ cover using heuristic search; generation of the plan of actions of the knowledge engineer ($Plan$) on the basis of the obtained fine coverage ($Cover$); generating a plan view (PR) based on coverage ($Cover$). For every stage necessary algorithms – including domain-specific heuristic function for reducing search space – were developed and experimentally studied.

4 General Characteristics of Basic Components of the Intelligent Software Environment of AT-TECHNOLOGY Workbench

The features of the implementation of the basic components of the intelligent software environment of the AT-TECHNOLOGY complex and the technology of the development of the IES based on it (1) were considered in detail in a plenty of works. Hence, we give here just a brief description of the software tools of the intelligent software environment, including the kernel, the user interface subsystem and the extension library for interaction with operational RUCs. The kernel implements all the basic functionality of automated support for the development of prototype IES, project file management, extension management, etc.

The technological KB is conventionally divided into an extension library that stores operational knowledge in the form of plug-ins that implement the relevant operational RUCs and the declarative part. The subsystem of the user interface has a convenient graphical interface, on the basis of which the RUC interacts with the knowledge engineer using screen forms (Fig. 1).

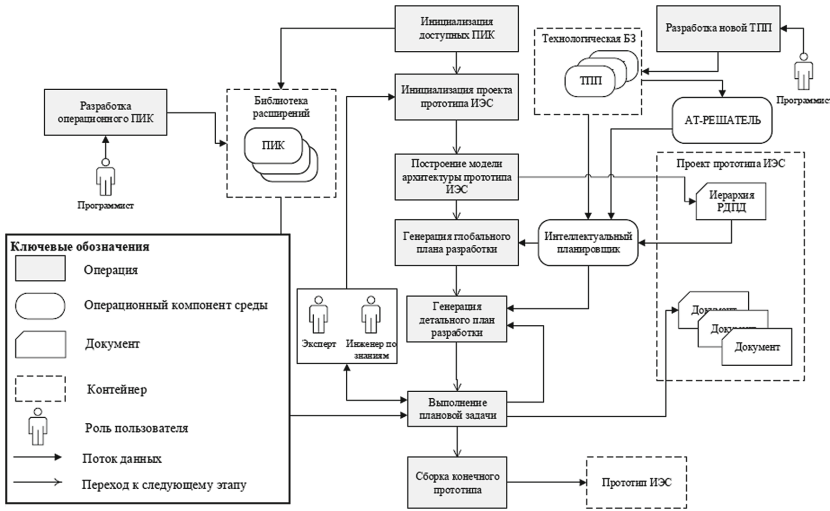


Fig. 1. The technology of the development of IES using intelligent software environment

The intelligent planner, being a part of the kernel, implements functionality related to planning the prototyping processes of IES: with the help of the preprocessor of the hierarchy of the EDFD, the pre-processing of the hierarchy of the EDFD is carried out by converting it into one generalized diagram of maximum detailing; the task of coverage of the detailed EDFD with the available SDP is performed with the help of a global plan generator that, on the basis of the technological KB and the constructed generalized EDFD, performs a fine coverage construction; generator of the detailed plan on the basis of the given coverage of the EDFD and the technological KB performs a detalization of each element of the coverage (thus forming a preliminary detailed plan); based on the analysis of available RUC, a detailed plan is formed in the plan interpretation component, where each task is related to a specific RUC and then with the help of the final plan building component, the required plan representation is generated.

By now, the technology of constructing prototypes of IES using a new version of an intelligent planner, three SDPs (including the SDP “Construction of dynamic IES”), as well as a set of operational and information RUC and other means of intelligent software environment of the AT-TECHNOLOGY workbench, has been experimentally studied.

5 Conclusion

Based on the data obtained as a result of joint testing of the intelligent planner and other components of the intelligent software environment of the AT-TECHNOLOGY workbench, it was shown that all the developed tools can be used quite effectively in the prototyping of applied IES. The software was integrated into the AT-TECHNOLOGY workbench, with the use of which several prototypes of a dynamic IES were developed, e.g., a prototype for managing medical forces and facilities for major road accidents and a prototype of dynamic IES for satellite network resources control.

Acknowledgements. The work was supported by the Russian Foundation for Basic Research (project №18-01-00457).

References

1. Geffner, H., Bonet, B.: A concise introduction to models and methods for automated planning. Online access: IEEE (Institute of Electrical and Electronics Engineers) IEEE Morgan & Claypool Synthesis eBooks Library, Morgan & Claypool (2013). https://books.google.ru/books?id=_KInIAEACAAJ
2. Ghallab, M., Nau, D., Traverso, P.: Automated Planning and Acting, 1st edn. Cambridge University Press, New York (2016)
3. Ghallab, M., Nau, D.S., Traverso, P.: Automated Planning - Theory and Practice. Elsevier, New York City (2004)
4. Giarratano, J.C., Riley, G.D.: Expert Systems: Principles and Programming, 4th edn. Course Technology, Boston (2004)
5. Meystel, A.M., Albus, J.S.: Intelligent Systems: Architecture, Design, and Control, 1st edn. Wiley, New York (2000)
6. Nau, D.S.: Current trends in automated planning. *AI Mag.* **28**(4), 43–58 (2007)
7. Rybina, G.V.: Theory and technology of construction of integrated expert systems. Monography. Nauchtehlitizdat, Moscow (2008)
8. Rybina, G.V.: Intelligent Systems: from A to Z. Monography Series in 3 Books. Knowledge-Based Systems. Integrated Expert Systems, vol. 1. Nauchtehlitizdat, Moscow (2014)
9. Rybina, G.V., Blokhin, Y.M.: Methods and means of intellectual planning: implementation of the management of process control in the construction of an integrated expert system. *Sci. Tech. Inf. Process.* **42**(6), 432–447 (2015)
10. Rybina, G.V., Blokhin, Y.M.: Use of intelligent planning for integrated expert systems development. In: IS 2016 - Proceedings of the 2016 IEEE 8th International Conference on Intelligent Systems, pp. 295–300 (2016)
11. Rybina, G.V., Blokhin, Y.M.: Methods and software of intelligent planning for integrated expert systems development. *Artif. Intell. Decis. Mak.* **1**, 12–28 (2018)
12. Rybina, G.V., Blokhin, Y.M.: Intelligent software environment for integrated expert systems development. In: Proceedings of the 2016 International Conference on Artificial Intelligence ICAI 2016. CSREA Press, USA (2016)
13. Schalkoff, R.: Intelligent Systems: Principles, Paradigms, and Pragmatics. Jones & Bartlett Learning (2011). <https://books.google.ru/books?id=80FXUtF5kRoC>