Intelligent software environment for integrated expert system designing and development

Galina V. Rybina,

doctor of technical science, prof.,

Yury M. Blokhin

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

1 Introduction

Today dynamic intelligent systems (DIS) are the most complex class of applied intelligent systems, including real time (RT) DIS. This is one of the most and currently one of the most relevant and in-demand classes of DIS are dynamic integrated expert systems (IES) that use dynamic domain and solving dynamic problems [10]. Analysis of the foreign and domestic level of research and development in the field of DIS, in particular, dynamic IES, has shown that when creating dynamic IES and relevant tools (IS), a large number of scientific and technological problems arise related to the specific features of building both separate components of the IES. And the organization of interaction of these components among themselves in the RV. As a whole, despite the lack of semantic unification of the terminology base, integrated DIS classifications, and their separate classes as well as the circle of general scientific and technological problems that hinder the wide application of DIS applications in strategically important subject domains, where the highest effect of using the temporal DIS occurs, have already accumulated [11-13].

Listed above problems substantially determine the high complexity of DIS development as dynamic integrated expert systems that are the most widespread and needed DIS class. Moreover, there is no universal complex method for solving the described problems (or a part of them) that implies the development of an integrated integral methodology and technology for creating such complicated systems at all lifecycle stages. Modern commercial software to support the construction of most DISs (G2, Rtworks, RTXPS, etc.) despite its power and versatility, is not able to solve the above problems in terms of integrated methodology fully [15] and others.

A new stage of developing the theory and technology of IES construction based on the problem oriented methodology is a valuable step towards IES development automation. Its main properties are stated in a few monographies [11-13]. Today, this is the basis that is used to create the intelligent applications and automated workstation of a knowledge engineer, namely, the AT TECHNOLOGY tool complex, on whose

basis several tens of applied IESs have been created, where a wide spectrum of models and methods of solving different unformalized and formalized problems is used in terms of integrated IES architecture.

A few generations of the AT-TECHNOLOGY workbench have proved efficiency in the development of more than 20 applied intelligent systems, including dynamic IES, which, in the context of a problem-oriented methodology, represent the further improvement of static IES related to working in dynamic environments and domains and solving dynamic problems (monitoring, diagnostics, planning, control, etc.). Thus in the architectures of dynamic IES there are components modeling the external environment and allowing to interact with hardware in the RT, and also to perform reasoning in the changing data and temporal knowledge. In general, this corresponds to specific results obtained by solving the scientific problems listed above (published in [10-13], etc.).

The modern version of the AT-TECHNOLOGY workbench enables automated support of IES development, using methods of intelligent planning and control of development processes [16,17] etc. As the conceptual basis of intellectual software technology is the concept of intelligent program environment model. Its complete formal description and methods for implementing individual components is given in [10].

The work is focused on the further development of methods and tools for the automated construction of dynamic IES using components of the intelligent software environment and taking into account the modern requirements of software engineering, described in detail in [5] and other works.

2 Intelligent planning methods and their usage for integrated expert systems development automation

The detailed analysis of modern methods, approaches and software tools used in the field of intelligent planning is given in [17], so here we consider only the most important theoretical and methodological aspects of this problem in the context of the goals and objectives of this paper.

The planning problem formulation described in modern works is usually based on the basic set of axioms for Labeled Transition System Σ, as [6,3]: finiteness; full observability; determinism; static; limited goals; plans linearity; implicit time; offline planning. These axioms impose significant restrictions on the formulation of the planning problem, and violation of some of these axioms leads to a complication of the planning task. The formal formulation of the planning problem is given in [3,6]. The most known approaches to planning were analyzed - graph planning, state space planning, transition to other problems, etc. [3,6,7].

The problem of planning of prototyping IES is quite well described in terms of states and transitions. It is necessary to point out that application of intelligent planning for the automated support processes of building intelligent systems is a poorly investigated area, and it is possible here to refer mainly to the experience gained in the creation of applied dynamic IES based on the problem-oriented methodology and

AT-TECHNOLOGY workbench, in particular, the development and use of educational and dynamic IES [14]. Let us consider the basic concepts of intellectual AT-TECHNOLOGY workbench software in more detail. So far, the main applications of intelligent planning are [1,3,4,6,17,18]: autonomous robots control; logistics and the resolution of extraordinary events; semantic web; automated tutoring; calibration of equipment; control of conveyor machines; resource-scheduling; resource allocation in computer systems.

The classical algorithm A* was chosen as the base for developing algorithms for generating global and detailed plans [17] used in the prototyping of PECs, was chosen, which is the simplest and sufficiently well-researched, which corresponds to modern world trends in the development of planners [9]. To reduce the search space, usually effective heuristic functions is used, which development in modern works on intellectual planning is given significant attention.

During the researches comparative analysis of universal heuristic functions used in planners implementations was carried out within the framework of these studies and using heuristics (relaxed heuristics, heuristics of the critical path, abstract heuristics, landmark) [2,3,8]: Blind; Relaxation-based maximum; Merge-and-shrink; Admissible Landmark; Relaxation-based additive; Relaxed plan heuristic; Casual graph heuristic; Context-enhanced additive heuristic; The Landmark Heuristic et al.

The comparison between different heuristic functions showed that even though the most powerful heuristic functions (for example, from the landmark class) show rather high efficiency, they do not give a fundamentally qualitative leap in the issues of computational complexity of solving the planning problem. Therefore, for specialized domains such as dynamic IES, it is preferable to use problem-oriented heuristic functions instead of universal ones. They can reduce search space size up to several orders, so in this work, the specialized heuristic function have been developed.

3 Implementation details of basic intelligent software environment components

Intelligent software environment takes significant place in the framework of the problem-oriented IES constructing methodology (basic points are reflected in [10]) and implements intelligent software support for the IES development processes. It is general concept of "intelligent environment". Complete formal description of the intellectual environment model and methods of the individual components implementation is presented in [10], so here only a brief description of the model in the form of quaternion is presented: $M_{AT} = <KB, K, P, TI>$, where KB is a technological knowledge base (KB) on the composition of the project, and typical design solutions used in development of IES. K - set of current contexts K_i, consisting of a set of objects from the KB, editing or implementing on the current control step. P – a special program - an intelligent planner that manages the development and IES testing process. TI - many tools TI_i, applied at various stages of IES development.

Intelligent planner is the main procedural (operational) component. It is defined as $P = <SK, AF, Pa, Pb, I, GP>$, SK here is the state of the current context, in which the

scheduler was activated. *AF* is a set of functional modules, a part of planner. *Pa* is a selection procedure for the current target based on the global development plan. *Pb* is a selection procedure for the best executive function module from the list of possible candidates. *I* - procedures to ensure the interface with the corresponding components of the AT-TECHNOLOGY workbench; GP - operating procedures for the IES global development plan.

The main technological knowledge unit is a standard design procedure (SDP), which can be represented as tuple $SDP_i = <C, L, T>$ where *C* - is the set of conditions under which the SDP can be implemented; *L* - script implementation described in the describing internal language actions of the *SDP*; *T* - set of parameters initialized by intelligent planner at SDP inclusion in the development plan of a IES prototype. Description of other intelligent software environment can be found, for example in [17].

Now let us state methods and approaches used in the implementation of the intellectual supportive environment for the development of IES model. The main components of this IES are the technological KB on the composition of IES project, SDP and RUC, and the intelligent planner managing the process of plans construction and implementation for the development of IES prototypes. These are the main purposes why it is necessary to use different types of knowledge in the process of developing a IES prototype: checking referential integrity of the project on the development of IES; automated construction of components diagrams; layout synthesis of IES prototype architecture; planning a series of steps to create a prototype of IES-specific features and tasks; determining a set of the most relevant sub-tasks for each of the stages (steps) in the development of IES prototype and others.

The main task of intelligent planner is a dynamic support knowledge engineer operations at all life cycle stages of building. Dynamic support is done by generating IES development plans for the current IES prototypes and allowing the specific plans execution (made either automatically or interactively). It should be noted that detailed plans and global IES prototyping generation and architecture model synthesis is based on the integration of IES with planning methods.

The proposed method is based on state-space planning which generates plan applied IES architecture model and the SDP set. The architecture model IES contains a set of elements that need to be realized and depending on the type of element and its content, different RUCs can be used for implementation, resulting in various components of the IES prototype project. Moreover, the use of knowledge from the technological KB (containing SDP and RUC) allows the implementation of several architecture mode components together with the application of the appropriate RUC. Four special algorithms have been developed to implement the proposed method. With the help of the first one, the model of the prototype architecture of the IES [10] is preprocessed by converting architecture model in the form of a hierarchy of extended data flow diagrams (RDPD) into one generalized diagram by recursive detailing of complex operations (example is shown in Fig.1. as a uncovered graph).

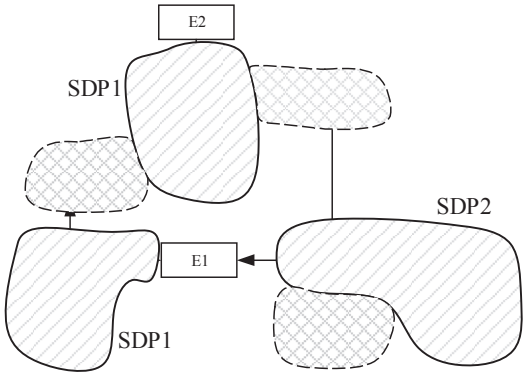

Fig. 1. Detailed coverage

The aim of the second algorithm is to search the *coarse coverage* of the generalized diagram (Fig.1, solid bound). In this paper, a multiple of SDP fragments is meant to cover a diagram, between elements of which a one-to-one correspondence with elements of a generalized diagram is established. Under a coarse coverage is meant a cover only by necessary fragments of a SDP.

It is performed in the following manner. At the first step, all the vertices of the diagram are marked with a number 0, and an empty set is also initialized, into which the activated SDPs will be added. Next, for each SDP from the technological KB, the possibility of covering this necessary SDP fragment on a fragment of a generalized diagram consisting of elements marked with zeros is checked. For the SDP in the set of activated SDPs, an instance of the corresponding SDP is added, and the covered elements are marked with the identifier of the corresponding instance. The process of generating coverage continues as long as there are SDPs in the technological KB, which can be covered with zeros of the generalized diagram. In this case, the same SDP can be activated several times in the form of different instances.

The third algorithm allows to generate detailed coverage (example is shown in Fig.1, where optional fragments bounded with dashed lines), which means covering all available SDP fragments from the set of activated SDPs. To do this an instance of the activated SDP is selected with attempt to cover the diagram elements marked with 0 using each of the optional SDP fragments, taking into account their links (data flows) with the required SDP fragment. The covered elements of the diagram are also marked with the identification number of the SDP instance. The algorithm is terminated when there are no optional SDP fragments that can extend the existing coverage.

The coverage generating with second and third algorithms are implemented with using heuristic state space search, based on the classical algorithm *A**. Generating of new states is associated with the coverage of the diagram by each new fragment (necessary and optional respectively). The fourth algorithm is designed to convert the

detailed coverage into a plan, with a set of planned tasks associated with the use of certain operational RUCs from each coverage fragment.

The reason for development of specialized algorithms is due to the violation of the axiom associated with setting constraints on the complexity of describing the goal state of the system. The generating of a global plan involves special search metric. To construct a coarse and accurate coverage, two specialized heuristic functions are used, which were developed on the basis of the experience of prototyping IES of different architectural typologies.

4 Conclusion

Now various methods of intelligent planning, algorithms and heuristic functions have been experimentally evaluated for the further development of the intelligent planner efficiency of AT-TECHNOLOGY workbench. As a result the following were completed: model of IES prototype development plan (decomposed to the global and detailed plans), as well as concretization of individual components of SDP model (scenario and conditions); the original method for IES prototype development plan generating based on heuristic search is developed as well as algorithms for its implementation, including specialized heuristic function.

On the example of development of two prototypes of dynamic IES for the Russian Center for Disaster Medicine "Zaschita" ("Management of medical forces and means for major road accidents" and "Management and monitoring of resources of the satellite communication system between regional centers"), software simulation of methods for development of particular components of dynamic IES prototypes of various architectural typology. Their design and development stages are characterized by high laboriousness and intellectual load on knowledge engineers.

The work was supported by the Russian Foundation for Basic Research support (project № 15-01-04696) and the MEPhI Academic Excellence Project (agreement with the Ministry of Education and Science of the Russian Federation of August 27, 2013, project no. 02.a03.21.0005).

References

1. Benton, J., Coles, A.J., Coles, A.: Temporal planning with preferences and time-dependent continuous costs. In: Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012, Atibaia, Sao Paulo, Brazil, June 25-19, 2012 (2012)
2. Bryce, D., Kambhampati, S.: A tutorial on planning graph based reachability heuristics. AI Magazine 28(1), 47–83 (2007)
3. Ghallab, M., Nau, D.S., Traverso, P.: Automated planning - theory and practice. Elsevier (2004)
4. Helmert, M., Gener, H.: Unifying the causal graph and additive heuristics. In: Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling, ICAPS 2008, Sydney, Australia, September 14-18, 2008. pp. 140– 147 (2008)

5. Lavrisheva, E.M.: Software Engineering of computer systems. Paradigms, technologies and CASE-tools of programming. Scien. dumka, Kiev (2013)
6. Nau,D.S.: Current trends in automated planning. AI Magazine 28 (4), P. 43–58 (2007)
7. Osipov, G.S.: Artificial intelligence methods. FIZMATLIT, Moscow (2011)
8. Penberthy, J.S., Weld, D.S.: UCPOP: A sound, complete, partial order planner for ADL. In: Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, October 25- 29, 1992. pp. 103–114 (1992)
9. Russell, S.J., Norvig, P.: Artificial Intelligence - A Modern Approach (3. internat. ed.). Pearson Education (2010)
10. Rybina, G.V.: Theory and technology of construction of integrated expert systems. Monography. Nauchtehlitizdat, Moscow (2008)
11. Rybina, G.V.: Intelligent systems: from A to Z. Monography series in 3 books. Vol. 1. Knowledge-based systems. Integrated expert systems. Nauchtehlitizdat, Moscow (2014)
12. Rybina, G.V.: Intelligent systems: from A to Z. Monography series in 3 books. Vol. 2. Intelligent dialogue systems. Dynamic intelligent systems. Nauchtehlitizdat, Moscow (2015)
13. Rybina, G.V.: Intelligent systems: from A to Z. Monography series in 3 books. Vol. 3. Problem-oriented intelligent systems. Tools for intelligent system developing. Dynamic intelligent systems. Nauchtehlitizdat, Moscow (2015)
14. Rybina, G.V., Blokhin, Y.M.: Distributed knowledge acquisition control with use of the intelligent program environment of the at-technology workbench. Commu- nications in Computer and Information Science pp. 150–159 (2014)
15. Rybina, G.V., Blokhin, Y.M.: Dynamic integrated expert systems: the evolution of the at-technology workbench. International Journal Information Content and Processing 1(2), 155–161 (2014)
16. Rybina, G.V., Blokhin, Y.M.: Use of intelligent program environment for construction of integrated expert systems. Life Science Journal 11(8), 287–291 (2014)
17. Rybina, G.V., Blokhin, Y.M.: Modern automated planning methods and tools and their use for control of process of integrated expert systems construction. Artificial intelligence and decision making (1), 75–93 (2015)
18. Tierney, K., Coles, A.J., Coles, A., Kroer, C., Britt, A.M., Jensen, R.M.: Automated planning for liner shipping fleet repositioning. In: Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012, Atibaia, Sao Paulo, Brazil, June 25-19, 2012. pp. 279–287 (2012)