Procedia Computer Science
Volume 88, 2016, Pages 205–210
7th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2016

Intelligent Programm Support for Dynamic Integrated Expert Systems Construction

Galina V. Rybina[1], Victor M. Rybin[1], Yury M. Blokhin[1], and Sergey S. Parondzhanov[1]

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

Abstract
The problems of intellectualization in the development process of integrated expert systems basing on the the problem-oriented methodology and the AT-TECHNOLOGY workbench are considered. The automation of dynamic integrated expert systems construciton is in focus. The intellgient programm environment and its basic components, including standard design procedure are reviewed. The detailed description of procedure for dynamic integrated expert system construction is given. The examples of applied integrated expert system prototypes developed with described procedure are listed.

Keywords: dynamic integrated expert system, problem-oriented methodology, typical design procedure, intelligent programm environment, intelligent planner, AT-TECHNOLOGY workbench

1 Introduction

Trends towards integration of research in different fields of artificial intelligence had most clearly manifested at the turn of the XX and the XXI centuries and made it necessary to combine semantically different objects, models, methodologies, concepts and technologies. As a result, new intelligent system architectures were emerged, in particular integrated expert systems (IES), i.e. systems with scalabale architecture and expansible functions [8, 9, 10, 11, 17, 3, 2].

Problem-oriented methodology for a wide class (static, dynamic, tutoring and others) of applied IES construction was developed by Rybina G.V. [8] from Department of Cybernetics National Research Nuclear University MEPhI. Problem-oriented methodology [8] is actively used and constantly develops: about 30 applied prototypes IES for diagnosis, control, design, planning and tutoring problems were constructed (detailed description can be found in [8, 9, 10, 11]). The core idea is based on the conceptual modeling of the IES architecture at all levels of the integration processes in the IES and focusing on the modeling specific types of unformalised tasks that are relevant to the technology of traditional expert systems. In the laboratory "Intelligent Systems and Technologies" of Department of Cybernetics in MEPhI several

Selection and peer-review under responsibility of the Scientific Programme Committee of BICA 2016

doi:10.1016/j.procs.2016.07.426

generations of software tools for the automated support of problem-oriented methodology were created [8, 9, 10, 11, 13, 15, 16, 14]. It is called AT-TECHNOLOGY workbench.

A large part of the problems is linked to the high complexity phases of design and implementation of the IES, as showed by practical experience of creating a series of static, dynamic, and tutoring IES through the use of problem-oriented methodology and AT-TECHNOLOGY workbench. Problem domain and human factor provide a significant impact on modelling process.

Designing of dynamic IES architecture is the most difficult phase, because in the dynamic IES important place is given to the integration of methods and means of temporal information acquisition, presentation and processing with the methods and means of the outside world simulation in real time. This leads to expansion of architecture (for example, intelligent control systems [16, 5, 7, 4]), built on the concept of dynamic IES relevant subsystems adequately reflecting all the processes and laws of functioning simulated systems [6, 16], as an integral phase of building dynamic IES.

Therefore, the need to develop intelligent software environment [8, 9] for the further development of problem-oriented methodology and AT-TECHNOLOGY workbench with the aim of creating intelligent technology to build specific classes of IES has become urgent. Now let us review basic features of problem-oriented methodology for IES construction, which is described in details in [8].

2 Features of Problem-Oriented Methodology for Integrated Expert Systems Construction

In the context of solving the modern IES construction problems (in particular for the management of complex discrete systems) problem-oriented methodology has the following properties [8, 9, 10, 11]: a powerful combination method of acquiring knowledge that supports the automated process of acquiring knowledge from the sources of knowledge of different typology (experts, database, text) is used to gain knowledge; generalized knowledge representation language designed for building models of problem areas in dynamic IES allows to represent temporal knowledge, based on a modified interval Allen logic [1] and time control logic, together with the basic knowledge, including those containing knowledge with uncertainty, imprecision and vagueness; supports the use of various output means (universal AT-SOLVER and a specialized temporal solver designed for dynamic tasks); in the context of enhanced functionality and principles of the components IES deep integration provides the possibility of implementing simulation techniques for modeling the external environment and how to interact with them; the high efficiency of the a large number of applied IES development, including dynamic areas of concern; instrumentally supported by a modern software such as workbench (AT-TECHNOLOGY).

Significant place in the framework of the problem-oriented methodology for IES constructing (basic points are reflected in [8]) is given to the methods and means of intelligent software support for the development processes, which form general concept of "intellectual environment". Complete formal description of the intellectual environment model and methods of the individual components implementation is presented in [8], so here only a brief description of the model in the form of quaternion is presented.

$$M_{AT} = \langle KB, K, P, TI \rangle \tag{1}$$

KB is a technological knowledge base (KB) on the composition of the project, and typical design solutions used in development of IES. $K = \{K_i\}, i = 1..m$ - 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. $TI = \{TI_i\}, i = 1..n$ - many tools TI_i, applied at various stages of IES development. A component of the KB is a declarative basis of intellectual support for the development of IES, acting as data storage in a given environment and defined as

$$KB = \langle WKB, CKB, PKB \rangle \quad (2)$$

WKB is a KB containing knowledge of the standard design procedures (SDP), describing the sequences and methods of using various tools to create applied IES and a sequence of steps for creating IES. CKB - is KB comprising knowledge about the use of SDP and reusable components (RUC), including fragments of previously created IES prototypes. PKB (optional) - is a KB containing specific knowledge used at various stages of creating IES prototype for solving problems that require innovative approaches.

The current context K_i is represented as set of $Ki = \langle KD, KP \rangle$. KD here is a declarative context for storing static declarative information about the structure of the project, the knowledge engineer and the current user. KP is a procedural context, which includes objects clearly affecting the further planner steps (LC system phase, currently edited or executable object, the current target, the current executor, the global development plan, etc.). The main procedural (operational) component is intelligent planner. This model generally describes it.

$$P = \langle SK, AF, Pa, Pb, I, GP \rangle \quad (3)$$

SK here is the state of the current context, in which the planner was activated. $AF = \{AF_i\}, i = 1..k$ is a set of functional modules AF_i, a part of planner. P_a 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. Each reusable component (RUC), involved in the development of an IES prototype, is represented by tuple:

$$RUC = \langle N, Arg, F, PINT, FN \rangle \quad (4)$$

N in this model is the name of the component, by which it is registered in the workbench. $Arg = \{Arg_i\}, i = 1..l$ - set of arguments containing current project database subtree serving the input parameters for the functions from the set. $F = \{F_i\}, i = 1..s$ - a variety of methods (RUC interfaces) for this component at the implementation level. $PINT$ - a set of other kinds of RUC interfaces, used by the methods of the RUC. $FN = \{FN_i\}, i = 1..v$ - set of functions names performed by this RUC. Any SDP can be represented as tuple

$$SDP = \langle C, L, T \rangle \quad (5)$$

, 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.

Today there are already implemented and used following SDPs: "Tutoring web-IES construction", "Distributed Knowledge Acquisition" and others. And currently researching SDP "Dynamic IES construction" is described below.

3 Description of Typical Design Procedure for Dynamic Integrated Expert Systems Construction

Described above SDP model is specified as $SDP_D = \langle C_D, L_D, T_D \rangle$ for dynamic IES construction, where C_D - is a set of conditions to initialize SDP_D realization; L_D - scenario of dynamic IES construction; T_D - a set of parameters initialized by intelligent planner when SDP_D is included into dynamic IES construction plan.

Conditions set C_D. Following set of conditions must be satisfied to include SDP_D into development plan:

- following lifecycle stage is system requirements analysis;
- in dynamic IES architecture model (as a hierarchy of EDFD) there is an element, describing simulation model presence (here, complex technical systems are simulated);
- there is at least one EDFD element, connected with solving of non-formalized problem.

Scenario L_D. There are following SDP_D scenario stages:

1. ***System equirment analisys stage***. Following actions are performed: automated knowledge acquisition based on temporal knowledge acquisition method [8], which performs direct acquisition process of knowledge, containing temporal references; simulation model development with help of specialized visual editor.
2. ***Design stage***. Here are performed following actions: forming of reasoning tools (a synergy of universal AT-SOLVER and Temporal Solver is currently supported); conversations of previously obtained knowledge field into knowledge base described in generalized knowledge representation language; explanation development; IES core components configuration.
3. ***Implementation stage***. During the final stage of dynamic IES prototype construction following steps are performed: simulation visual representation development; development of components for communication with user with language for dialogue scenario description;integration with external systems (DB, applied software modules, etc.); aggregate integration of IES components.

Parameters set T_D. In case of including SDP_D into development plan there are two parameters included into current context: one identifies currently executed SDP, the other one - current scenario step.

Now, using [12], let us review features of dynamic IES prototyping with SDP_D, which functions are supported by a set of operational RUC in each IES construction stage. Let us assume, that knowledge engineer already built dynamic IES architecture model as a EDFD hierarchy. With help of AT-SOLVER in automated mode initial architecture layout of IES prototype is formed as it is shown in Fig. 1. Global plan is also generated with intelligent planner core. Detailed development plans generation is performed iteratively, and following selection and execution of the most priority task is performed with following modification of architecture layout. The process is looped, until all tasks are not done. Detailed plan is regenerated in it is necessary.

Because it is usually assumed, that IES construction planning problem actions are performed with deterministic result, so deterministic planning approach is used as base [12]. It is performed

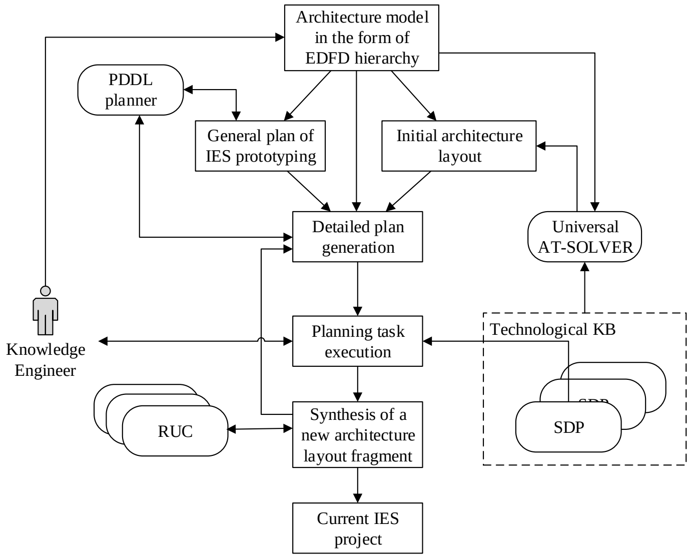

Figure 1: Dynamic IES construction planning scheme

in project state space, formed by sets of possible values of project parameters. Built EDFD hierarchy is analyzed with intelligent planner in following manner. Elements which have to be implemented are recognized, for example simulation model, DB fragments, non-formalized tasks, etc. These elements represent big tasks and precedence relation between them is built, in particular with use of PDDL-planners.

Initial architecture layout generating and global IES development plan generating is performed once after EDFD hierarchy is build. Then detail development plan is generated iteratively with help of intelligent planner after each architecture layout fragment adding. Detailed plan is visualized with visualization component, and knowledge engineer can initialize specific plan task execution. Each plan task is performed with use of appropriate RUC of AT-TECHNOLOGY workbench.

Samples of another complex SDPs, for example, connected with tutoring IES development are described in [8, 9, 10, 11]. The difficulties of the tutoring IES development technology are caused by supporting two different work modes DesignTime, oriented to work with teachers (course/discipline ontology creating processes, different typed training im-pacts creating, etc.) and RunTime, for working with students (current student model building processes, including psychological model, etc.).

4 Conclusion

Currently, an experimental software study of the current version of the intelligent planner during educational IES prototyping on various courses is carried out. This study is carried out in particular, for the collective development of IES prototypes with limited resources.

Research and development related to the use of described SDP "Dynamic IES construciton" with intelligent software environment for two dynamic IES prototypes ("Management of medical forces and resources for major traffic accidents" and "Resource management for satellite communications system between regional centers") and tutoring IES prototypes for different courses/disciplines were succesfully performed.

4.1 Acknowledgments

This work was supported by the Russian Foundation for Basic Research under grant no. 15-01-04696 and Competitiveness Growth Program of the Federal Autonomous Educational Institution of Higher Professional Education National Research Nuclear University MEPhI (Moscow Engineering Physics Institute).

References

[1] James Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] B.E. Fedunov. Intelligent systems for the core of anthropocentric objects and its modeling. pages 394–400, 2012.

[3] Crina Grosan and Ajith Abraham. *Intelligent Systems - A Modern Approach*, volume 17 of *Intelligent Systems Reference Library*. Springer, 2011.

[4] I. M. Makarov, V. M. Lokhin, S. V. Manko, and M. P. Romanov. *Artificial intelligence and intelligent control systems*. Nauka, Moscow, 2006.

[5] G. S. Osipov. *Artificial intelligence methods*. FIZMATLIT, Moscow, 2011.

[6] M. Pidd. *Computer simulation in Management Science*. Wiley, 5th ed. Chichester, 2004.

[7] V.M. Rybin and S.S. Parondzhanov. Using of dynamic integrated expert systems for intelligent control. *International Journal of Applied Engineering Research*, 10(13):33202–33205, 2015.

[8] G. V. Rybina. *Theory and technology of construction of integrated expert systems. Monography*. Nauchtehlitizdat, Moscow, 2008.

[9] G. V. Rybina. *Intelligent systems: from A to Z. Monography series in 3 books. Vol. 1. Knowledge-based systems. Integrated expert systems*. Nauchtehlitizdat, Moscow, 2014.

[10] G. V. Rybina. *Intelligent systems: from A to Z. Monography series in 3 books. Vol. 2. Intelligent dialogue systems. Dynamic intelligent systems*. Nauchtehlitizdat, Moscow, 2015.

[11] G. V. Rybina. *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.

[12] G. V. Rybina and Y. M. Blokhin. 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.

[13] G. V. Rybina and A. O. Deineko. Distributed knowledge acquisition for the automatic construction of integrated expert systems. *Scientific and Technical Information Processing*, 38:1–7, 2011.

[14] G.V. Rybina and Y.M. Blokhin. Methods and means of intellectual planning: Implementation of the management of process control in the construction of an integrated expert system. *Scientific and Technical Information Processing*, 42(6):432–447, 2015.

[15] G.V. Rybina and A.V. Mozgachev. The use of temporal inferences in dynamic integrated expert systems. *Scientific and Technical Information Processing*, 41(6):390–399, 2014.

[16] G.V. Rybina, V.M. Rybin, S.S. Parondzhanov, and S.T.H. Aung. Some aspects of simulation application in dynamic integrated expert systems. *Life Science Journal*, 11(SPEC. ISSUE 8):144–149, 2014.

[17] A. R. Tyler. *Expert systems research trends*. Nova Science Publishers, New York, 2007.