Use of intelligent program environment for construction of integrated expert systems

Galina Valentinovna Rybina, Yury Mikhaylovich Blokhin

National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoye shosse, 31, Moscow, 115409, Russian Federation

Abstract. Development of the integrated expert systems with the task-oriented methodology and the problems of the intellectualization of AT-TECHNOLOGY workbench are reviewed. Intelligent planning methods applied for tutoring IES development processes are described with usage of the intelligent planner, reusable components, typical design procedures, and another intelligent program environment components. Some peculiarities of designing tutoring web-oriented integrated expert systems (web-IES) using intelligent development support tools functioning inside the AT-TECHNOLOGY workbench are described. As an example, typical design procedure «Tutoring web-IES building» schemes for two different modes are presented.

[Rybina G.V., Blokhin Y.M. **Use of intelligent program environment for construction of integrated expert systems.** *Life Sci J* 2014;11(8s):287-291] (ISSN:1097-8135). http://www.lifesciencesite.com. 64

Keywords: tutoring integrated expert systems, task-oriented methodology, AT-TECHNOLOGY workbench, intelligent program environment, intelligent planner, typical design procedures, reusable components

Introduction

Support problems of development processes of intelligent and technological integrated expert systems (IES) with power functionality and scalable architecture are getting more and more significant and topical. For the first time these problems were reviewed in IES [1] development task-oriented methodology and in AT-TECHNOLOGY workbench which supports this methodology and represents knowledge engineer workstation [1].

The experience collected in development of multiple applied IES [1], tutoring IES development and usage in particular [2] have shown that the most of IES development problems are related to high complexity of projecting and implementation stages, and the problem domain has significant influence on organization and specificity of these lifecycle stages. The human factor also remains significant enough, because it brings labor rising and extends lead time, and in most cases the IES development technology specificity doesn't allow to use traditional programming methods.

Therefore a significant place in the task-oriented methodology was assigned to methods and instruments of intelligent program support of development processes. To all of them a common term «intelligent program environment» is applied (common methodology provisions are described in monograph [1] and another papers, i.e. [2,3]). Full formal intelligent environment description and some components implementation methods are presented in [1], therefore we shall present here in brief a description of the model $M_{AT} = \langle KB, K, P, TI \rangle$, where KB – a *technological knowledge base* (KB) about project composition and typical project solutions used in IES development; $K = \{K_i\}, i = 1,...,m$ – a current context array K_i, consisting of objects from KB that are editing or running during the current control step; P – a special program – *intelligent planner*, which controls IES development and testing processes; $TI = \{TI_i\}, i = 1,...,n$ – a variety of instruments TI_i which are used in different IES development stages.

KB is a development intelligent supporting declarative base, which is acting as an environment informational storage and is defined as $KB = \langle WKB, CKB, PKB \rangle$, where WKB is a knowledge base with the knowledge about *typical design procedures* (TPP), which are describing sequences and variants of using the instruments for the applied IES development as well as the IES development stage sequences; CKB – a knowledge base containing the knowledge about TPP and reusable components (PIK) usage, including the knowledge about early developed IES prototypes; PKB (optionally) – the specific knowledge used in a variety of IES development stages in some specific tasks, i.e. the tasks which need a nonstandard development approach.

Current context K_i is defined as $K_i = \langle KD, KP \rangle$, where KD – the declarative context, designed for storing static declarative information about the project, a knowledge engineer and a current user; KP – the procedural context including the objects which influence directly the further planner steps: a lifecycle stage, a currently edited or executed object, a current goal, a current executor, a global development plan etc.

The main procedural (operational) component – intelligent planner P model is defined as $P = \langle SK, AF, Pa, Pb, I, GP \rangle$, where SK – the current context state, which invokes the planner; $AF = \{AF_i\}, i = 1,...,k$ – the variety of AF_i functional modules which are parts of the planner; Pa – the current goal choosing procedure in accordance with the global development plan; Pb – the procedure of selecting the best executor from the functional components; I – the procedures providing the appropriate AT-TECHNOLOGY component interfaces; GP – the procedures of interacting with the global IES prototype development plan.

Thus, the base components of the IES development support intelligent program environment are the tools of knowledge engineer intelligent planning (i.e. intelligent planner).

It is necessary to point out that modern intelligent planning methods are widely discussed in many proceedings and papers, in particular [4-6], and they have a wide range of application in the following fields: autonomous robots control, i.e. [6]; web-services composition (i.e. getting more functions with combination of some services for cooperative execution) [7]; automated tutoring (in particular for generating individual learning routes [8]); hardware calibration with resources and time limits [9]; resource scheduling [4,10,11], resource management [12], manufacturing processes [13]; in computation resources management in complex distributed systems and real-time systems [14].

Usually, using intelligent planning methods involves integration with the knowledge of representation methods, temporal reasoning, restriction satisfaction methods, genetic algorithms etc.

However, the concept «intelligent planning» still doesn't have any unique definitions across the world. In Russian proceedings it is mostly called «intelligent planning» (i.e. [5]), but in Europeans and Americans – «automated planning» [6]. In both cases a generating process based on a computer plan is related. In the present work the term «intelligent planning» is used.

So far a few generations of intelligent planners for AT-TECHNOLOGY[1] workbench have been developed with intelligent planning models and methods using in IES [1,3]. A new research step related with further evolution of base intelligent environment components (including new version of intelligent planner with modern intelligent planning methods, new operational systems abilities and graphical libraries) is the subject of the present work.

Common features of the implementation models and methods of the base components of the intelligent IES development support environment

Consider shortly methods and approaches used in intelligent IES development supporting environment implementation, base components: technologic knowledge base about project composition, TPP and PIK; intelligent planner, that controls generating and execution IES prototypes development plan processes. List common goals, which achievement needs usage of different types of knowledge in IES prototype development process: checking project referential integrity; automated component diagram creation; IES prototype architecture model generating; IES prototype development steps sequence planning considering solving tasks specificity and features; most topical subtasks selecting in each development stage and etc.

IES prototype development plan generating is artificial intelligence task, because it involves different knowledge about typical tasks solving models and methods [1], about IES projecting and developing, about external integration with databases and other applications and etc. Thus IES development project based on task-oriented methodology is called aggregate of knowledge and data about solving task (stored in some formats on physical storage), which are base for intelligent planner supported IES prototyping processes.

It should be noted that current intelligent planner implementation bases on HTN [15] and flexible reasoning used in IES approaches hybridization, that allows using declarative knowledge (in this case – some kind of If/Then rules) about development process.

For base IES prototype development control processes description, which implementation needs bringing some types of knowledge and intelligent planner usage, we will give main models short description according to [1].

IES prototype modes is defined as: $PRJ = \langle PN, KB, Solver, Pd, PDFD, PPIK, PCOMP \rangle$, where PN – project name, KB – IES prototype knowledge base, $Solver$ – IES prototype solver; PD – project data, i.e. different typed information (knowledge, data, parameters, texts, etc.), used by intelligent planner in IES prototype development processes and for final prototype generating, and besides expert interviewing protocols, lexis dictionaries, knowledge field fragments, knowledge bases in different knowledge representation languages, solving task type, and some kind of technical information (current user profile, knowledge engineer name who created project, project start and end dates, etc.) can be referred to base project data; $PDFD$ – extended informational-logical IES prototype architecture model as extended data flow diagrams hierarchy (EDFD), which is one of the most important project data, because its structure defines prototype composition and its functional abilities (EDFD is constructed with only one relation – decomposition, i.e. higher level operation is detailed with EDFD of lower level, and only one top level EDFD can exist in a project); $PPIK = \{PPIK_i\}, i = 1,...,n$ – variety of PIK; $PCOMP = \{PCOMP_i\}, i = 1,...,l$ – different IES subsystems aggregate developed with AT-TECHNOLOGY workbench and external tools.

IES prototype development plan is defined as $PL = \langle S, PP, A, R, D \rangle$, where S – set of IES prototype development stages, which are usually understood as IES prototype lifecycle stages; PP – a set of TPP; A – a set of elementary tasks which must be executed with knowledge engineer for successful IES prototype development; R – relation «included» defining including of tasks from A into TPP from a set PP; D – relation «is topical in stage» characterizing tasks from a set A or TPP from a set P belong to exact IES prototype development stage.

Any TPP model is defined as: $PP_i = \langle C, L, T \rangle$, where C – a set of conditions, which satisfaction makes possible TPP invocation; L – execution scenario described with internal TPP actions description language; T – a set of parameters are initialized with the intelligent planner in case of including TPP into an IES prototype development plan.

Every PIK, involved in IES prototype development is defined as $PIK = \langle N, Arg, F, PINT, FN \rangle$, where N – component name which is used for registration in workbench; $Arg = \{Arg_i\}, i = 1,...,l$ – a set of arguments comprising current project database sub trees, which are used as input parameters for functions execution from a set $F = \{F_i\}, i = 1,...,s$ – a set of methods (PIK interfaces) in implementation layer; $PINT$ – a set of another PIK interface naming, which are used in current PIK, $FN = \{FN_i\}, i = 1,...,v$ – a set of function naming, which are executing with current PIK

Thus, the main task of the intelligent planner is dynamic support of the knowledge engineer during every lifecycle stage by IES prototype development plan building and plan execution providing.

Use of typical design procedures and of reusable components in integrated IES development process

The main algorithm element used during development plan generation process of the IES prototype is TPP. By TPP a set of elementary instructions (steps) is meant, which are traditionally executed by a knowledge engineer at every development lifecycle stage. The intelligent planner of the AT-TECHNOLOGY workbench has knowledge about all available TPPs, and basing on this knowledge it forms a set of tasks for any IES prototype development (accordingly to a current lifecycle stage). Then, basing on special requirements specified at the system requirements analysis stage, the planner decomposes the plan into smaller tasks (subtasks).

All workbench TPPs are classified in the following manner: task type independent TPPs (for example, “knowledge acquisition from database”), task type dependent TPPs (for example forming tutoring IES components), TPPs related with PIK, i.e. procedures, that contain knowledge about PIK lifecycle from its configuring up to including it into the IES prototype model. TPPs of the last type also contain knowledge about tasks solved with this PIK and its necessary configurations.

The common architecture of the AT-TECHNOLOGY workbench is built in such manner, that all functionality is distributed between the components registered in the workbench and acting under intelligent development environment. In other words, these components are reusable components of the workbench, and they are developed in accordance with some workbench rules [1].

There are two different types of PIK used in the current base AT-TECHNOLOGY workbench version – procedural and informational components. In the first one the components provide capabilities for execution either actions with non-typical results, i.e. results that are not stored in some special storage (repository) as the results of previous developments, or actions, that require user interaction (for example, editing the ER-scheme or viewing the expert interviewing protocol). In the second one the components provide capabilities for executing actions which result in the information that has been collected earlier and is stored in the repository (knowledge, data, schemes, structures etc.) with further copying of this information into the current project and preprocessing if needed (i.e. copying of created earlier ER-diagram or typical diagram analysis). Special storages (repositories) are used for PIK of the second type. They collect different types of data which is used in further development processes.

Some peculiarities of the TPP for tutoring IES construction

In the base AT-TECHNOLOGY workbench many TPPs of the first and second types are implemented and used, in particular: the TPP for combined knowledge acquisition, the TPP for database projecting, the TPP for configuring IES prototype components, the TPP for creating hyper-text tutorials etc. There are TPP related with distributed knowledge acquisition from different knowledge sources, dynamic IES development TPP and the most complicated TPP for tutoring IES construction [2] in the experimental stage. The difficulties of the tutoring IES development technology are related with supporting two different work modes – DesignTime, oriented to work with teachers (course/discipline ontology creating processes, different typed training impacts creating, etc.) and Runtime, for working with students (current student model building processes, including psychological model, etc.).

In Fig. 1 the execution scheme of TPP «Tutoring web-IES construction» in DesignTime mode is presented, and in Fig.2. - in RunTime mode.

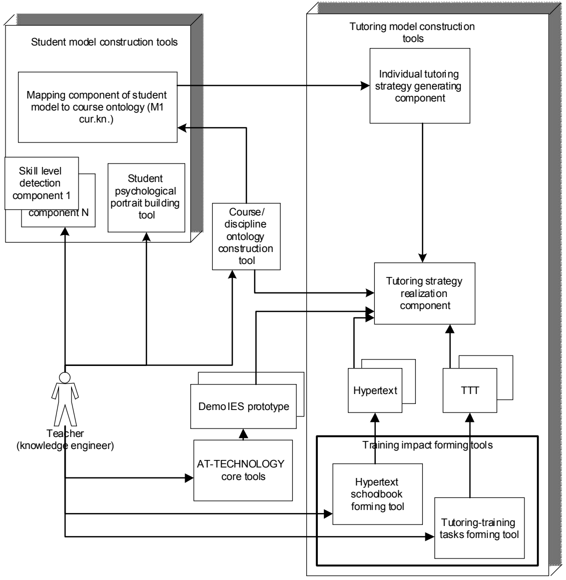
Student model construction tools
Mapping component of student model to course ontology (M1 cur.kn.)
Skill level detection component 1
component N
Student psychological portrait building tool
Course/ discipline ontology construction tool
Tutoring model construction tools
Individual tutoring strategy generating component
Tutoring strategy realization component
Demo IES prototype
Hypertext
TTT
Teacher (knowledge engineer)
AT-TECHNOLOGY core tools
Training impact forming tools
Hypertext schodbook forming tool
Tutoring-training tasks forming tool

Fig. 1. Execution scheme of the TPP «Tutoring web-IES construction» in DesignTime mode for a teacher (or a knowledge engineer)

As shown in Fig. 1., in the RunTime mode the following AT-TECHNOLOGY workbench instruments are used: training impact building tools (hypertext schoolbook, tutoring-training tasks), base core workbench tools for IES prototype construction, student psychological portrait builder tool, tools for course/discipline ontology building, individual learning strategy former, learning strategy realization component, different skills level detection components, component for mapping a current student model to course/discipline ontology.

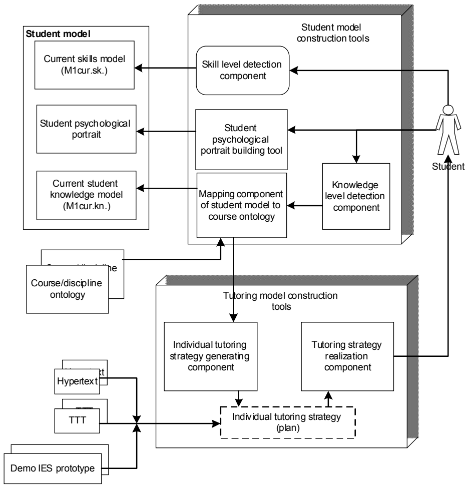
Student model
Current skills model (M1cur.sk.)
Student psychological portrait
Current student knowledge model (M1cur.kn.)
Student model construction tools
Skill level detection component
Student psychological portrait building tool
Mapping component of student model to course ontology
Knowledge level detection component
Student
Course/discipline ontology
Tutoring model construction tools
Individual tutoring strategy generating component
Tutoring strategy realization component
Individual tutoring strategy (plan)
Hypertext
TTT
Demo IES prototype

Fig.2. Execution scheme of the TPP «Tutoring web-IES construction» in the RunTime mode for students

In the scheme of Fig.2., is shown that student model construction (M1cur.kn., M1cur.sk., psychological portrait components) is directly connected to the tools for construction and realization of the tutoring model, as well as to the component for mapping a current student model to course/discipline ontology. The mapping component also is connected to the individual tutoring strategy forming component. The peculiarity of tutoring web-IES developed with AT-TECHNOLOGY workbench, is a presence of some components of skill level detection and of the component for building psychological student portrait (as an aggregate of personal characteristics which are collected in results of psychological testing).

Consider this TPP in context of topicality of intelligent development technology usage for tutoring IES. Schemes shown in Fig.1 and Fig.2. of the TPP «Tutoring web-IES construction» conveniently show a big amount of repeating routine operations which must be performed by a knowledge engineer during the projecting lifecycle stage (DesignTime mode) and maintenance (RunTime mode) of the tutoring web-IES for certain courses/disciplines.

The most difficult and complicated stage is the construction of the «Training with IES» training impact, which represents the comprehensive problem of the applied IES development for a certain problem domain with using of base AT-TECHNOLOGY workbench tools. For example, almost all the courses of the «Intelligent systems and technologies» specialization usage require some knowledge of engineering methods. These methods are presented as non-formalized tasks and non-formalized methods such as «System analysis of the problem domain about applicability of the ES technology», «Knowledge representation formalism choosing», «Development tools choosing» and other tasks containing expert knowledge [1]. The aggregate of the listed non-formalized tasks and their logical relations are the base for the problem domain construction. The problem domain is constructed with the knowledge representation language used in the AT-TECHNOLOGY workbench.

Conclusion

Tutoring IES development and its using in learning process at the NRSU MEPHI beginning from 2008 have proved that the tools of intelligent support of complex and multifunctional IES development processes allow to decrease specific IES designing time, to reduce intellectual work of knowledge engineers and teachers and to provide a flexible use of the ac-cumulated course/discipline tutoring experience.

The collected experience has shown how difficult it is to develop and prototype tutoring web-IES as well as to maintenance and monitor their informational data and software. Usually, without appropriate resources all the work must be done by teachers and theirs students. Therefore further researches must be focused on creating a powerful intelligent technology for supporting intelligent tutoring systems in educational field.

Acknowledgements

This work is executed in NRNU MEPhI, supported by RFBR(project12-01-00467)

Corresponding Author:

Dr.Rybina Galina Valentinovna
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
Kashirskoye shosse, 31, Moscow, 115409, Russian Federation

References

1. Rybina, G.V., 2008. Theory and Technology of Integrated Expert Systems Construction. Moscow: Publishing house “Nauchtekhlitizdat”, pp: 482.
2. Rybina, G.V., 2011. Intelligent tutoring systems based on integrated expert systems: development and usage experience. Information-measuring and Control Systems, 10: 4-16.
3. Rybina, G.V., 2011. Intelligent technology of integrated expert system construction implementation models and methods. Devices and Systems. Control, monitoring, diagnostics, 10: 27-37.
4. Burkov, V.N., N.A. Korgin and D.A. Novikov, 2009. Theory of organizational system management introduction, Moscow: Publishing house “Librokom”.
5. Osipov, G.S., 2011. Artificial intelligence methods, Moscow: Publishing house “FIZMATLIT”.
6. Nau, D.S., 2007. Current trends in automated planning. AI Magazine, 4(28): 43–58.
7. Zou, G., Y. Chen, Y. Xu, R. Huang and Y. Xiang, 2012. Towards Automated Choreographing of Web Services Using Planning. Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI Press, pp: 178-184.
8. Garrido, A., L. Morales and I. Serina, 2012. Using AI Planning to Enhance E-Learning Processes. Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, AAAI Press, pp: 47-55.
9. Parkinson, S., A. Longstaff, A. Crampton and P. Gregory, 2012. The Application of Automated Planning to Machine Tool Calibration. Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, AAAI Press, pp: 216-224.
10. Goffman, I.D., D.A. Inishev, A.A. Lipatov, I.E. Smirnov and K.E. Smirnov, 2008. Topical tasks of uncertain calendar planning Time-EX development. 11th national conference on artificial intelligence KII-2008, Moscow: Publishing house “LENAND”.
11. Beaudry, E., F. Kabanza and F. Michaud, 2010. Planning with Concurrency under Resources and Time Uncertainty. Proceedings of the 19th European Conference on Artificial Intelligence ECAI 2010, IOS Press, pp: 217-222.
12. Coles, A., 2012. Opportunistic Branched Plans to Maximise Utility in the Presence of Resource Uncertainty. Proceedings of the 20th European Conference on Artificial Intelligence ECAI 2012, IOS Press, pp: 252-257.
13. Gladkov, L.A. and M.A. Alexandrov, 2012. Operational planning task solving with modified genetical algorithm used with uncertaint logical controller. 13th national conference on artificial intelligence KII-2012, Belgorod: Publishing house BGTU.
14. Kolesov, N.V., M.V. Tolmacheva and P.V. Yuhta, 2011. Computation planning timecost and complexity assesment in distributed systems. Proceedings of 4th Russian multiconference, Taganrok: Publishing house TTI YFU.
15. Sacerodi, E.D., 1975. The nonlinear nature of plans. IJCAI'75 Proceedings of the 4th international joint conference on Artificial intelligence, Morgan Kaufmann Publishers Inc., pp: 206-214.

5/20/2014