Intelligent Software Environment for Integrated Expert Systems Development

Galina V. Rybina¹, Yuri M. Blokhin¹
¹Department of Cybernetics, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, 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 integrated expert systems architecture model generating are described with usage of the intelligent planner, reusable components, typical design procedures, and other intelligent program environment components.

Keywords: integrated expert systems, problem-oriented methodology, AT-TECHNOLOGY workbench, intelligent software environment model, automated planning, automated planner.

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. It created new classes of problems and new architectures of intelligent systems. The systems based on knowledge or expert systems (ES) (they were originally intended to support the decision formalized problems (UF-problems)), are an essential part of a significant number of static and dynamic intelligent systems, as the analysis of development experience [1-6] has shown. At the same time, in conjunction with ES, methods of applied mathematics, soft computing and versatile means of control data are used for solving tasks with practical complexity and importance. It leads to the necessity of combining ES into a single architecture with such diverse components as: a database comprising engineering, production and management information; software packages with the developed calculating, modeling and graphics tools; educating systems, including built on technology training and learning systems; simulation modeling systems for dynamic applications, etc.

A new class of intelligent systems emerged - integrated expert systems (IES). Their scalable architecture is represented in the form of ES + K. K - some software that provides decision support for the formalized part of the problem, and function of finding solutions to the UF-tasks relevant to the components of the poorly structured problem falls on the ES.

To address IES as an independent object of research, the IES terminological basis was formed and the basic definition of the IES as a systematically organized set of components ES and K was introduced. Their structure and nature of the interaction determine the type of the IES relevant and solvable problems - its functionality. The concept of "static IES" and "dynamic IES" was also introduced [1]. Automated task-oriented methodology [1] was created for the construction of IES, with powerful functionality and scalable architecture. It is actively used and constantly develops. 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 UF-tasks that are relevant to the technology of traditional ES. In the laboratory "Intelligent Systems and Technologies" Department of Cybernetics National Research Nuclear University MEPhI was created several generations of software tools such as Workbench (AT-TECHNOLOGY workbench) for the automated support for task-oriented methodology [1-4,7].

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 educating IES through the use of task-oriented methodology and AT-TECHNOLOGY workbench [1-4,8]. Specifics of a particular area of concern and the human factor provide a significant impact. Therefore, the need to develop intelligent software environment and its basic component - intelligent planner [1,2,9] for the further development of task-oriented methodology and AT-TECHNOLOGY workbench with the aim of creating intelligent technology to build specific classes of IES has become urgent.

To date, multiple versions of intelligent planners for the AT-TECHNOLOGY workbench were created. They were developed by combining predictive models and methods of planning methods used in IES [1,2,9-14].

This paper point of issue is a new research phase and obtained results relating to the further development of intelligent planner and other components of the intellectual environment. The purpose for the new research is to increase the degree of automatic (intellectualization) planning and project management for the creation of a broad class of IES.
2 Some aspects of current research in the field of intelligent planning

Today automatic generation of plans by a software and hardware system is often meant as the intelligent planning, but the term “intelligent planning” has no clear definition. For example, the term “intelligent planning” is more commonly used in Russian literature [15], while “automated planning” is used in English [16]. However, in both cases the plan generation process done by the computer is meant. Accordingly, plan is a glimpse of future behavior in the context of intelligent planning. In particular, the plan is usually a set of steps with some restrictions (e.g., temporal) for the execution of some agent or agents. [16].

Software system that generates plans based on a formal description of the environment, the initial state of the environment and assigned to the planner purpose is meant under planner in practical application. In some works, planner is called the agent [15]; in other cases, the planner also contains the designer carrying out the allocation of resources for the implementation of building plans.

A significant number of methods, approaches, formalisms, etc. was developed by now in the field of intelligent planning. Among them should be highlighted: planning with propositional logic; planning in a space plans; planning in space of conditions; planning as constraint satisfaction problems; planning on the basis of precedents; broadcast to the other problem; temporal planning; planning in non-deterministic and probabilistic areas; hierarchical planning (HTN-formalism), and others. Detailed reviews can be found in [15-17], and others, as well as in the works of authors [11-14].

Methods of the intelligent planning are widely used in a number of applications. The most popular fields of intelligent planning applications are: management of autonomous robots [16,18]; semantic web and web services composition [19]; computer-aided learning (in particular for the construction of individual education plans [10,20]); calibration of equipment [21]; control of conveyor machines [22]; resource-scheduling [23]; resource allocation in computing systems [24] logistics [25], the automation of software development [26] and others.

PDDL (Planning Domain Definition Language) developed under the McDermott’s leadership is the current norm in the field of the intelligent planning with the three main versions. We can detect the following trends in the context of development and use of various intelligent planning formalisms and related languages: the unification and standardization of planning languages on the basis of PDDL; the emergence of task-specific planning languages; a shift in emphasis towards nondeterministic research planning (RDDL languages and PPDDL); the emergence of complex mechanisms to meet the preferences and limitations, and others.

However, in general, 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 IES based on the task-oriented methodology and AT-TECHNOLOGY workbench, in particular, the development and use of educational and dynamic IES [10,28,29]. Let us consider the basic concepts of intellectual AT-TECHNOLOGY workbench software in more detail.

3 Model of intellectual software environment and its components

Significant place in the framework of the task-oriented IES constructing methodology (basic points are reflected in [1]) is given to the methods and means of intelligent software support for the development processes. It is general concept of "intellectual environment". Complete formal description of the intellectual environment model and methods of the individual components implementation is presented in [1], so here only a brief description of the model in the form of quaternion is presented.

\[ M_{AT} = <KB, K, P, TI> \]  

\( KB \) 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

\[ KB = <WKB, CKB, PKB> \] 

\( WKB \) 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 re-used 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 \( K_i = <KD, KP> \). \( KD \) 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.

\[ P = \langle SK, AF, Pa, Pb, I, GP \rangle \]  

(3)

\( SK \) here is the state of the current context, in which the scheduler was activated. \( AF = \{AF_i\}, i = 1..k \) is a set of functional modules \( AF_i \), 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.

Any SDP can be represented as triples

\[ SDP_i = \langle C, L, T \rangle \]  

(4)

, 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. Each RUC, involved in the development of an IES prototype, is represented by the tuple:

\[ RUC = \langle N, Arg, F, PINT, FN \rangle \]  

(5)

\( N \) in this model is the name of the component, by which it is registered in the complex. \( Arg = \{Arg_i\}, i = 1..l \) - 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.

Prototypes IES model of the functioning process is represented as triples:

\[ M_1 = \langle Sc, C, Cl \rangle \]  

(6)

\( Sc \) is a scenario of IES prototype work. \( C \) - set of IES prototype subsystems that can be divided into 2 categories (standard subsystem (RUC) from the RUC repository and prototype's IES subsystem implemented by developers). \( Cl \) is a ratio of "control transfer ", describing the procedure for interaction between the IES prototype's subsystems.

Let us briefly examine the 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.

4 Development plan construction for the applied IES prototype

The task of creating a plan for developing an applied IES prototype is a complete task from the field of artificial intelligence, because it requires involvement of a variety of knowledge about models and methods for solving typical problems [1], a technology on design and development of IES, on how to integrate with external databases, packaged applications and other programs. Therefore, a project to develop IES-based task-oriented methodology and data on the problem being solved is stored in some format on physical carrier body of knowledge. Based on this data the intelligent planner is running the process of prototyping for applied IES.

It should be noted that the implementation of the current intelligent planner version [14] is a hybridization of approaches based on the use of HTN-formalism and flexible mechanisms to find solutions used in the IES. It allows the use of a declarative way of describing knowledge of the development, in this case the production-type knowledge representation language [1].

The implementation of managing the IES prototype development is a basic process, and requires the use of certain types of knowledge and use of intelligent planner. A brief description of some of the basic models for this process in accordance with [1] is given.

The model of the IES prototype is presented as a seven:

\[ PRJ = \langle PN, KB, Solver, PD, PDFD, PPIK, PCOMP \rangle \]  

(7)

\( PN \) - project name, \( KB \) - IES prototypes KB, \( Solver \) - output machine (agents) for the IES prototype. \( PD \) - project data, i.e. the different types of information (knowledge, data, individual parameters, text, etc.), used by the intelligent planner in the process of developing a IES prototype and to generate the finished prototype, wherein the main project data include records of interviewing experts, lexical dictionary, fragments of the knowledge field, KB in the different
IES prototypes model development plan is presented by the cinque:

\[ PL = <S, PP, A, R, P, D> \]  \hspace{1cm} (8)

\( S \) is the set of IES prototype development stage, usually meaning life cycle (LC) stages of the IES prototyping. \( PP \) – a set of SDP. \( A \) - number of tasks whose implementation by the knowledge engineer is necessary for the successful development of a IES prototype; \( R \) - relation 'part of the' between the elements of the \( A \), plurality from set \( A \). \( P \) - consequence relation between the objectives. \( D \) - ratio that determines the feasibility of the task \( A \), from set \( A \) using a specific SDP from a \( PP \) variety. Hierarchical tasks network in the form of \( TN = <A, R> \) can be defined for the \( PL \) plan using the HTN-formalism.

Plan of the IES prototyping can be decomposed into global and detailed. The global plan is a

\[ PLG = \{AG, PG, CA\} \]  \hspace{1cm} (9)

, where \( AG \subset A; \) \( PG \subset P \), defined on the set of \( AG \); \( CA \) - relationship "fit architecture models component", between the elements of the \( AG \) components plurality and the architecture of the IES prototype PDFD model, described as ADFD hierarchy. It should be noted that each component of the IES architecture model prototypes development is a complex process, and can be detailed as a complete specific tasks network. Detailed plan \( PLD \) is a global \( PLG \) plan detailed using HTN-based formalism, and is defined as \( PLG = \{A, P, TN\} \).

IES prototype architecture model refers to the set of interconnected RUC and other subsystems that solve the problems described at the stage of user’s system requirements analysis with the help of information and logical model.

Here is a general description of the IES prototype development plan synthesis problem. Assume \( KU \) - the set of all possible \( K \) contexts. Then the task of planning the IES prototype development can be represented as

\[ PlanningTask = <K, AU, KG> \]  \hspace{1cm} (10)

\( AU \) is the set of all available actions of knowledge engineer on the \( PRJ \) project, \( K \) - is a \( KU \) element with full description, \( KG \) - is an element of \( KU \) and represents (perhaps not complete) description of the target context of the project. Ordered set \( DevPlan = ap_1 ... ap_n \), where each \( ap_i \in AU \), is the planning problem’s solution, if superposition of operations \( ap_n (... (ap_2 (ap_1 (K))) ... ) \) eventually leads to a \( KG \) goal state. Previously described model of the \( PL \) plan is the \( DevPlan \) models development based on HTN-formalism and taking into account additional information.

Thus, the main task of intelligent planner is a dynamic support knowledge engineer operations at all life cycle stages of building prototypes as shown in Fig. 1. 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.

Since the task of planning the IES prototype development usually implies an unambiguous outcome of the planned tasks implementation, determinate planning has been chosen as the main approach [15,16]. Of course, this approach does not explicitly take into account developments related to the risks and failures of the project due to the human factor, but this is offset by the restructuring plan strategies in the event of deviations from the plan. The environment for the scheduler is a description of the IES prototype project and entirely foreseeable.

Planning is performed in a state space of the project [16] (in the early studies the concept of "design space" was used). State space is formed by a plurality of project parameter’s possible values. This approach was chosen because of its popularity and the large number of tools that can be effectively integrated into AT-TECHNOLOGY workbench intelligent planner.

Plan to build an IES prototype is a complete project development task, and it is a reason why elements of the plan have to be tied to the time. According to this [30] temporal approach with an explicit modeling of time was chosen, as it is planned to adapt the intelligent planner under the team development management in the further research. It is also another argument in favor of temporal planning. Described above formalism and language PDDL 2.1 is used as a basis for combining these approaches. It allows taking the time into account and using real variables.
Fig. 1. The process of building IES prototypes based on the integration of planning with the IES methods

5 Features of model components realization for the intelligent software environment

Third generation of the AT-TECHNOLOGY workbench [1-4,7,14] is fairly complex modern software designed to solve problems associated with the design, development and maintenance of static and dynamic IES. The basis of the implementation of the workbench current version is approach based on technology using dynamic extensions (plugins), providing a modular and easy interchangeability of additional developer tools (knowledge engineer), implemented in the form of RUC complex.

Intelligent environment is implemented within the framework of the AT-TECHNOLOGY workbench architecture to support the development of IES in the form of several software modules (intelligent planner project manager, etc.) and data storage (project repository, the technological KB, etc.).

Software implementation of the intelligent planner current version is in the form of a few blocks. The main ones are: the architecture model synthesis component, plan visualizer, interaction with the knowledge engineer component, a component generation development plan for a prototype IES and others. These are implementation features of the most important components [12-14].

Plan visualizer. It is designed to visualize current plans and to initiate a specific task of a plan for the knowledge engineer.

Component generation development plan for a prototype IES. Provides a generation process for the generalized plans (based on the architecture IES model) and detailed plans (using the current IES architecture model and generalized plan). These processes are implemented using interaction with external PDDL-planners through component integration.

The PDDL-planners integration component. To solve the planning problem, intelligent planner uses external PDDL-planners, including LAMA, COLIN [31].

Methods of intellectual planning are used to solve described above tasks with intelligent scheduler planning. These methods are: deterministic classical planning in state-space model with the trivial time (to generate a global plan) and temporal planning with the action cost (to generate a detailed plan). These methods are implemented with the help of external programs - PDDL-planners, and the integration is based on file sharing. For PDDL-planners the task is created in the PDDL language version 2.1 [30] with additional enhancements that are further addressed in a separate process.

Separation to the global and detailed plan allows effective optimization to the intelligent planner operations, retaining its response time, as the number of states even for a small scheduling problem can be quite high [32]. The task of planning for PDDL-planner must be submitted in the form of quaternary.

$$P = \langle AP, IP, OP, GP \rangle$$

In this model $AP$ is a set of variables determining the state; $IP$ - the initial state; $OP$ - the set of operators; $GP$ - target condition. Then converting of the planning tasks PlanningTask with intelligent planner is made as follows: a set of $AP$ PDDL- predicates is formed by $PRJ$, $KU$ and the ratio $R$ (to determine the hierarchical relationship between tasks); set of true predicate and metric value $IP$ is formed from $PRJ$ and the current context $K$; from $AU$ and $PRJ$ a lot $OP$ PDDL-operators (action) are generated; from the target context $KG$ and $PRJ$ a lot of truth for the target state PDDL-predicates are generated. It should be noted that the set of $OP$ operators in the PDDL language is given implicitly through the use of typing and additional predicates.

The architecture model generation component. Produced generation of initial architecture model based on IES architecture model prototypes (represented as ADFD hierarchy), the generation of architectural model fragments by performing the objectives of the plan, as well as the merger of the new piece of architecture with the current layout. Generation of the initial model is done taking into account the specifics of the problem area that the IES prototype is developed for. That’s why generic AT-SOLVER [1] is used to build the initial model architecture. It allows generating fragments of architecture model based on the subject area knowledge.

The configuration component. Configuration management is carried out, and it determines the choice of PDDL-planner, a directory for intermediate files, etc. Configurations are stored as XML-files, and component includes a graphic interface for editing configurations.

Interactive tasks in the development are transmitted to the means of AT-TECHNOLOGY interface, and the results are obtained from the interaction of a knowledge engineer. It is done with the help of the integration component and a variety of RUC. In addition, cooperation with RUC AT-TECHNOLOGY workbench by the transmit / receive messages is realized, and in order to highlight certain characteristics of the project and monitor the implementation of interactive objectives of the plan (by sending RUC messages) perform a number of internal project processing functions is performed to develop a prototype of the IES.

In the process of architecture model generation AT-SOLVER transmits instructions for specific kinds of tasks to the integration component with a variety of RUC. Specially developed language is used for a description of instructions. It is a subset of XML, because as a reporting format used by the components of the AT-TECHNOLOGY workbench, XML was selected.

Thus, the use of intelligent planning in managing the process of building IES prototypes based on ask-oriented methodology and AT-TECHNOLOGY workbench allows for the following functionality: the construction and implementation of the IES prototype’s development plan at all stages of life cycle with intelligent planner; dynamic assistance to the knowledge engineer in the construction of current IES prototype based on the SDP and RUC knowledge; synthesis of IES prototype architecture and its components on the basis of enhanced information and logical IES architecture model; IES prototype analysis by using knowledge of the models and methods for solving typical problems; issuing recommendations and explanations to the knowledge engineer.

The amount of technological KB of the AT-TECHNOLOGY workbench currently stands at about 120 rules that are used to implement SDP at all life cycle stages of the IES development for problem areas associated with the diagnosis and design. In addition the development of IES prototypes uses about 70 RUC, of which 20 realize the capability of information RUC based on a single repository, and other 50 implement possible procedural RUC.

6 Conclusions

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 intelligent software environment for building software applications for two dynamic IES prototypes are presented ("Management of medical forces and resources for major traffic accidents" and "Resource management for satellite communications system between regional centers").

The work was done with the Russian Foundation for Basic Research support (project № 15-01-04696).

7 References


