SOME ASPECTS OF TEMPORAL REASONING INTEGRATION WITH SIMULATION MODELING FOR DYNAMIC INTEGRATED EXPERT SYSTEMS CONSTRUCTION USING AT-TECHNOLOGY WORKBENCH

Galina Rybina, Victor Rybin

Abstract: *The scientific and technological problems of constructing dynamic integrated expert systems and the approaches to their solution are discussed. The new stages of the development of a problem-oriented methodology and AT-TECHNOLOGY workbench for constructing integrated expert systems in the context of solving dynamic integrated expert systems are considered. The theoretical and technical issues concerned with the representation and processing of temporal knowledge and simulation modeling are of special interest. The features of the implementation of temporal reasoning software and simulation modeling software for the dynamic version of the AT-TECHNOLOGY workbench are described.*

Keywords: *dynamic integrated expert systems, problem-oriented methodology, AT-TECHNOLOGY workbench, time representation, interval Allen logic, temporal solver, integration, simulation modeling*

ACM Classification Keywords: *software and its engineering, real-time systems software, software notations and tools, context specific languages, development frameworks and environments, software development techniques*

Introduction

Dynamic integrated expert systems (IESs) and technologies are actively applied practically in all fields of social human activities [Рыбина, 2008; Рыбина, 2015]. As a whole, despite the lack of semantic unification of the terminology base, dynamic IESs classifications, and their separate classes as well as the circle of general scientific and technological problems have already accumulated [Рыбина и др., 2013; Рыбина и др., 2015]:

1. Difficulties in obtaining temporal knowledge (i.e., knowledge, where time is taken as the essence of the subject domain) using different sources (experts, texts, databases, etc.) for dynamic representation of the subject domain (SD).
2. The complexity of the development of formalisms for dynamic presentation of the subject domain, which is determined by the variable composition of its essences, a change of input data

coming from the external sources (the external world model) in time, and the need for structuring, storing, and analyzing data that vary in time.

3. The complexity of solving dynamic problems, which concerns the implementation of the concurrent temporal reasoning about several different asynchronous processes (tasks), limited resources (time and memory), and changes in the composition of knowledge and data during problem solution.

4. Problems of simulation modeling the external world (surroundings) and its different states in real time at all stages of design and development of the system up to the startup.

5. The high cost of foreign software to support the development and maintenance of dynamic IESs of different classes, as well as the practical absence of domestic facilities.

6. The necessity for special software and hardware facilities to connect to the external world (sensors, controllers, etc.).

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 tools to support the construction of most dynamic intelligent systems (G2, Rtworks, RTXPS, etc.) despite its power and versatility, is not able to solve the above problems in terms of integrated methodology fully [Рыбина, 2015; Рыбина и др., 2015; Rybina et al., 2014a].

A considerable step towards generating such methodology can be a new stage of developing the theory and technology of IESs construction based on the problem-oriented methodology, whose main statements were offered by G.V. Rybina in the 1990s [Рыбина, 2008]. Today, this is the basis that is used to create the intelligent programs and automated workstation of a knowledge engineer, namely, the AT-TECHNOLOGY workbench, 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 [Рыбина, 2008; Рыбина, 2015].

The analysis of the above problems of constructing dynamic IESs that are central in the present paper shows that the closely related methods to obtain, present and process the temporal knowledge with simulation modeling the external world are least studied in terms of the system approach and the integrated methodology development. The ability to present the temporal relationships between the events that occur in a system and their use during the search for a problem solution allow one to reduce the search space considerably, which influences the functioning speed of dynamic IESs as a whole. In the above-mentioned commercial tools (G2, Rtworks, RTXPS, etc.), the approaches to time reflection, much less temporal knowledge, are rather simple. They are practically not used in the solution of dynamic problems [Рыбина и др., 2015].

Thus, there is a need to create the models, methods, and software tools that execute procedures for temporal inference and simulation modeling in dynamic IESs. These models, methods, and software tools must be integrated in terms of integrated methodology and technology, roles that the problem-oriented methodology of constructing the IESs and the supporting AT-TECHNOLOGY workbench play. Moreover, the problems of integration of simulation modeling technology with dynamic IESs got the most complete development under problem-oriented methodology and supporting software tools (AT-TECHNOLOGY workbench) [Рыбина, 2015; Рыбина и др., 2014].

The focus of this article which is a continuation of the researches described in [Рыбина и др., 2015; Rybina et al., 2014a; Рыбина и др., 2014; Rybina et al., 2015] is technological and applied aspects connected with the expansion of the architecture of the dynamic version of the AT-TECHNOLOGY workbench by integrating the subsystem simulation modeling of the external world and the combined functioning of this subsystem with temporal solver and other basic workbench components in the development of dynamic IESs prototypes.

Some aspects of temporal inference in dynamic IESs

The representation of temporal relationships and analysis of different models and methods of time presentation is of special interest to modern research in the dynamic intelligent systems area. Papers [Shoham, 1987; Shoham et al., 1988; Еремеев и др., 2004; Spranger, 2002; Allen, 1991], as well as time control papers [Осипов, 2008; Осипов, 2011], point linear time model papers [Еремеев и др., 2004; Еремеев и др., 2009], interval temporal logic papers [Еремеев и др., 2012; Allen, 1983; Плесневич, 1999], branching time papers [Еремеев, 2006; Ladkin et al., 1990], and [Рыбина, 2015; Рыбина и др., 2013], where arguments justify the use of the Allen interval logic [Allen, 1983] and time management [Осипов, 2008] in the dynamic version of the AT-TECHNOLOGY workbench, were devoted to this question.

As mentioned above, the modified Allen logic based on the classical logic [Allen, 1983] and the time control logic [Осипов, 2008] were selected for the dynamic presentation of the subject domains (SDs) in terms of analysis and experimental investigation. Let us explain the idea of the proposed method. To represent temporal knowledge in terms of the basic knowledge representation language (KRL) of the AT-TECHNOLOGY workbench [Рыбина, 2008], a generalized KRL (GKRL) [Рыбина и др., 2015] for dynamic IESs has been developed. It allows one to present temporal knowledge based on the modified Allen logic and the timecontrol logic together with basic knowledge, including knowledge with vagueness, inaccuracy, and carelessness. The main elements of the basic KPL are objects and rules.

The objects correspond to SD essences, which are described as follows [Рыбина, 2008]: an object (IO, NameO, L), where IO is the sequence number of the object; NameO is the object name; L is the list of attributes of the following type: Attribute (IA, NameA, Type), where IA is the sequence number of an attribute; NameA is the attribute name; Type is the type of the form: Type (IT, NameT, U), where IT is the number of the attribute type; NameT is the name of the attribute type; U is the set of possible values of the attribute (the list of certain values of the attribute, or the range of the values in which the maximum and minimum of the attribute or the membership function are specified).

In the GKPL, the event appears as an object of the SD in the form of one with the main attribute, namely, the origin condition. The type of the given attribute is a logical expression, whose value is an ordinary logical expression that connects the attributes of other SD objects that are not temporal primitives (this is a new type of element of the U set).

The validity of the value of the attribute origin condition at time T shows that the given event occurred at the time instant T. Moreover, the event also has the attribute the number of origins, which is an integer and describes the number of SD observation times. In a similar manner, the temporal interval appears as an object with two main attributes, viz., beginning and end conditions. Both the attributes are of the type “logical expression.” The truth of the attribute value of the beginning condition at the time instant T1 shows that the given interval started at T1, while the truth of the attribute value of the end condition at T2 shows that the given interval terminates at T2, i.e., the interpretation of the given interval on the temporal axis is the section [T1, T2], where T1 is always less than T2. The interval also has two more attributes, viz., the number of origins (like events) and the duration (an integer that describes the observation duration of the interval in the SD).

The rule in the basic KPL [Рыбина, 2008] is in the form of (IR, Ins, Cons), where IR is the sequence number of the rule; Ins, a rule parcel containing the list of the “attribute–value” pairs relating to each other by the logic relationships of conjunction and disjunction; Cons is a rule action, which contains the list of attributes with the assigned values. The parcel of the temporal production rule in the GKPL with such an approach involves (apart from ordinary components) a local model of the event development described by events, temporal intervals (taking durations into account), their origin frequencies, and the relationships between them. The application of the right part (signification) of the rule in this case can be executed only under the correspondence of this local model to the current event development in the SD. Thus, the use of the modified Allen logic allows one to describe the temporal dependences between the objects of the SD directly inside the rules; tracing the displays of these relationships results in solutions that take the current event development in the SD into account.

Let us now consider the features of time control, which requires introduction of new rules into the KPL. On the one hand, such rules should provide the rapid reaction on certain (usually, urgent) events of the SD (“rules–reactions”). On the other hand, there is a need to watch certain cycles in operation (“periodic rules”). As a whole, the “rules–reactions” corresponds to the modified Allen logic, being rules that contain the elementary formula conditions in the parcel. These formulas make up a single temporal object (event or interval). To present the “periodic rules,” the new attribute TYPE, which is able to take one of two values, viz., “Ordinary” and Periodic, is introduced in the KPL rule. The parcels Ins of periodic rules add an extra condition, which is the execution period.

Temporal reasoning software (temporal solver)

As a result of the research work there was developed temporal reasoning software (temporal solver) deeply integrated with the all-purpose solver (AT-SOLVER) within the AT-TECHNOLOGY workbench [Рыбина, 2015; Рыбина и др., 2015; Rybina et al., 2014a]. Temporal solver is included into dynamic extension of the AT-TECHNOLOGY workbench. That allows dynamic IESs successfully operate both in static and dynamic problem domains. The temporal solver, which is one of the new components of the dynamic version of the AT-TECHNOLOGY workbench executes the solution of two problems, which are the construction of the interpretation of the event development model in the SD and the signification of the temporal part of the production rules during functioning according to the problem formulation of the temporal inference on rules and functional requirements.

Let us briefly consider the functions of the modules and blocks of the temporal solver [Рыбина и др., 2015; Rybina et al., 2014a]. *The module for the interpretation of the event development model* provides the construction of initial interpretation and its modification on each step and includes several blocks:

The event and interval loading block loads the events and intervals into the internal representation of event descriptions and intervals of the SD;

The event and interval identification block checks the correspondence between the data that enter at each operation step and the event and interval origin condition;

The non-standard situation solution block provides the solution of the conflicts between the input data and the current interpretation (e.g., observation of the end of an interval up to its beginning).

The production-rule processing module is intended for the loading and signification of the temporal parts of rules according to the current interpretation of the event development model.

The block for integration with other components allows the program interface to interact with the rest of the components of the AT-TECHNOLOGY workbench [Рыбина, 2014].

Some aspects of the application of simulation modeling software in the construction of dynamic IESs

In the context of the use of the problem-oriented methodology for constructing IESs [Рыбина, 2008], the additional functionality of described tasks entails a significant change of the IES architecture as all basic components of static IES are practically modified, especially, knowledge base and reasoning tools, and two new subsystems are added—subsystem modeling the external world (environment) and subsystem interfacing with the physical equipment, as well as the technology of constructing dynamic IESs is significantly changed. The subsystem interfacing with the external environment is necessary to obtain a constant data stream from external equipment and sensors, and the subsystem modeling the external world (environment) is intended to simulate the data stream at all stages of the life cycle of dynamic IES development [Рыбина, 2015; Рыбина и др., 2014].

In the context of this work, the subject of discussion is the subsystem modeling the external world, because the data that is transferred to working memory by the subsystem uses temporal and all-purpose solvers of the AT- TECHNOLOGY workbench [[Рыбина, 2008] Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. Монография. – М.: Научтехлитиздат, 2008. − 482с.; Рыбина, 2014; Рыбина, 2015] to realize a deduction and to obtain recommendations. Basing on these objectives for computer simulation of the complex engineering systems (CES) and complex engineering and organizational systems (CEOS) behavior in time the simulation modeling concept using RAO-approach [Емельянов и др., 1998], which implements the process-oriented approach to construct simulation models (SM) achieved the most development and application. As the expansion experience of the AT-TECHNOLOGY workbench architecture by specialized tools in the form of the simulation modeling subsystem of the external world which is realized on the basis principles of the RAO-approach [Рыбина и др., 2014; Рыбина, 2014; Rybina et al., 2014b] has shown, this way was quite effective for deep integration of all components of the dynamic IES nucleus and combined functioning of the simulation modeling subsystem with the temporal solver, all-purpose AT-SOLVER and other basic components of the dynamic version of AT-TECHNOLOGY workbench.

The basic principles of the simulation modeling subsystem implementation based on RAO-approach and task-oriented methodology requirements are considered in more detail. In the architecture of the simulation modeling subsystem, the functionality of the developed tools is divided between two global modules [Рыбина, 2015; Рыбина и др., 2014; Rybina et al., 2015] – the “SM development module” whose tasks are to support the development process and debugging of SM and other functions requiring the visual interface, and the “SM computation module” ensuring the computation of the conditions of SM in each time step (cycle) of the functioning process of the dynamic IES.

The development of a powerful full-featured high-level language to describe the SM and the creation of a corresponding compiler for that language are the unifying conceptual framework for the two basic modules. To implement this approach, at the first stage of the researches, formalism RAO [Емельянов и др., 1998] is used as a language to describe the SM, the basic version of which is given in [37]. In the future, based on the analysis of current requirements to design models of CES / CEOS to create dynamic IESs developed a special language “RAOAT” including new conceptual changes associated with object -oriented language and significant technological expansions due to the addition of new instructions and data storage structures [Рыбина, 2015; Рыбина и др., 2014; Rybina et al., 2015; Рыбина, 2014].

Composition and structure of the “SM development module” and the “SM computation module” functioning in the structure of the current version of the simulation modeling subsystem (dynamic version of the AT-TECHNOLOGY workbench) are briefly considered bellow. General architecture, composition and structure are of the basic components of the current version of simulation subsystem, detailed description of which is given below [Рыбина, 2015; Рыбина и др., 2014; Rybina et al., 2015; Рыбина, 2014].

Visual Objects Editor: The component “Visual objects editor” allows you to create objects, setting their properties and attributes, as well as establish relationships between the model objects in graphical mode. Knowledge engineer and / or specialist on simulation modeling, operating with the models editor, designs SM on graphic canvas containing a predetermined number of various objects, at that the ability to create, delete, copy objects and set relationships between them is ensured. The values of a particular set of properties are able to change for each object. The created visual representation of SM and properties of all objects are stored in the memory of the editor, as well as in a separate text file, which is further processed by the RAOAT language compiler. This visual tool allows you to load a saved SM to update the visual presentation and insert the changes manually to obtained code which is described in RAOAT language.

Models Synthesis Component: The models synthesis component, interacting with the visual objects editor by processing the stored collections of objects, generates a description of SM in RAOAT language in XML format which is passed to the RAOAT language compiler.

Component of Visualizing SM: Knowledge engineer, if necessary, with the support of the “Animation frames and displayed rules editor” selects the description of the model in RAOAT language and makes animation frames for corresponding objects, and the tool “Visualizer" based on the values of resource parameters, descriptions of animation frames and displayed rules ensures drawing the animation frames.

RAOAT Language Compiler: The obtained description of SM in RAOAT language is passed to the input of the “SM computation module” where the compilation from RAOAT language to C# language occurs and the further interpretation and run the developed model. The kernel of the “SM computation module” is the “RAOAT language compiler” which structure has a standard form for the syntax-driven three-pass compiler. This compiler consists of an analyzer which includes components of the lexical, syntactic and semantic analysis, and synthesis component including a generation component of output code. It should be noted that the availability of such objects in the RAOAT language, as irregular events and temporary resources, requires the time coordination with each object of the model. Each object has its own internal timer showing within the modeling time scale, how much time obtains an object. In addition, this timer is associated with the total time of an activity of the SM for a corresponding pause during transferring and obtaining the data from the working memory of the temporal solver.

Supporting Component of Computing the SM States: The “Supporting component of computing the SM states” ensures the generation of a discrete modeling time, as well as the generation of the control actions used to start or stop the activity of SM in the form of messages on each discrete time step. Computation of the new state of SM is based on the state at the previous time step and an allowance of executed operations.

Some aspects of the interactions between the temporal solver, all-purpose AT-SOLVER and the simulation modeling subsystem

The important feature of the temporal solver is the close interaction with the all-purpose AT-SOLVER and the subsystem of the simulation modeling of the external environment (external world), which is an obligatory component of any dynamic IES. The temporal solver, as well as the subsystem of simulation modeling, acts on the times and process of the interaction between the temporal solver, while the subsystem of simulation modeling is carried out by data and command exchange in the asynchronous mode.

Figure 1 shows the chart of the interactions between the temporal solver, all-purpose AT-SOLVER, and the simulation modeling subsystem. These interactions are provided by joint functioning support facilities. Moreover, the components interact with the total working memory. It should be noted that the interaction is carried out in two modes: the development of the applied dynamic IESs (including the adjustment of a series of IES prototypes) and functioning of the final prototype of the dynamic IESs. The first mode that is needed for dynamic IES construction is the first that was considered in the present paper.

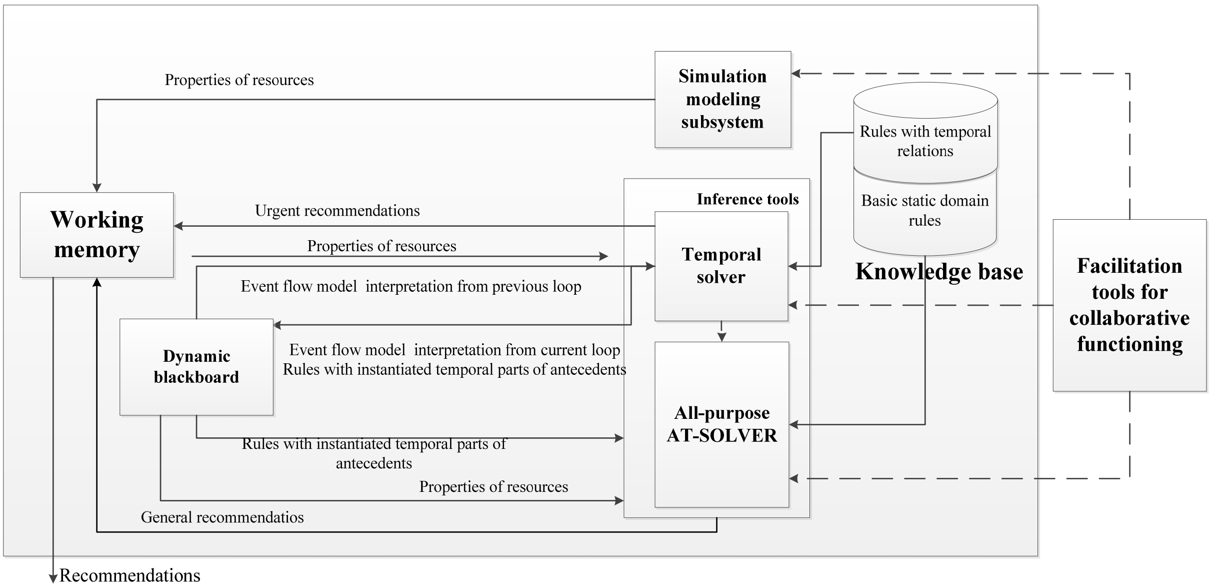

Fig. 1. Interaction diagram of temporal solver, all-purpose solver, and simulation modeling subsystem

The interaction between the components begins after the support facilities receive a message about the start of the adjustment of the developed prototype of the dynamic IES. At the initial stage, this is the configuration of components, including the setting of the duration of the operation cycle for the simulation modeling subsystem, indicating the database for the temporal solver and the AT-SOLVER.

The joint functioning support facilities provide the synchronization of component operation by sending messages with start or stop commands. The selected objects of the SD, whose collection of attributes describes the system state, are presented in the working memory. The KB contains the temporal rules that are necessary to solve the formulated problem, as well as to describe the events and intervals. As a result of temporal inference on rules, the system state changes, i.e., the attributes of the working memory objects vary according to the solved problem. A synchronous interaction would mean that the simulation modeling subsystem goes to the standby mode up to the inference completion by the temporal solver following data transmission (from the subsystem to the solver). Otherwise, the asynchronous interaction is said to be the ability to continue the operation of the subsystem without waiting for the temporal inference. Asynchronous interaction allows higher productivity due to the use of the time of the processing of general situations by the temporal solver to execute the next modeling cycle. Note that similar asynchronous interactions are applied in real practice, when it is impossible to react to an event immediately. This is the reason that the temporal solver and subsystem are synchronous.

The functioning of the simulation modeling subsystem and the inference facilities is an asynchronous process that is executed in parallel. The functioning of the temporal solver and ATSOLVER is an synchronous process that is executed sequentially. The interaction between the components of the AT-TECHNOLOGY complex is a very difficult process that requires the development of models, methods, and software facilities to support interactions. The functions of the modules and blocks are the following:

The configuration block carries out the component configuration. It sets the duration of the cycle of the discrete model time and the assignment of the names for the simulation modeling subsystem objects, temporal solver, and AT-SOLVER.

The model time generation block counts the cycles of the discrete model time according to the cycle duration specified by the configuration block. The working memory scanning block observes the changes in the working memory.

The control effect calculation block implements the target function of the interaction model. As a result of the block operation, the target component and the control effect, which should be set, are defined.

The control effect generation block forms the control effect as the message to the certain component. The interface module of the message exchange with the components processes the input messages and sends control effects. For joint functioning support facilities, the special adjustment tools allow one to emulate the combined work of the components of AT-TECHNOLOGY in both the stepbystep and realtime modes.

The use of these tools allows the study of the operation of the main components of the dynamic version of AT-TECHNOLOGY in the fullest manner.

Conclusion

These experimental investigations showed the advantages of the developed software tools compared to similar ones according to such criteria as the KPL power, operation speed, and reduction of the lead time of dynamic IESs. Verification of the performance and efficiency of these tools was done by developing a set of basic components, which is the minimum that is needed for dynamic IES operation.

Acknowledgements

This work was supported in part by the Russian Foundation for Basic Research under Grant No. 15-01-04696

Bibliography

[Allen, 1983] Allen J. Maintaining knowledge about temporal intervals // Communications of the ACM. 1983. N.26 (11). P.832-843.

[Allen, 1991] Allen. J. Time and time again: The many ways to represent time // Journal of Intelligent Information Systems. 1991. N. 6(4). P. 341-356.

[Ladkin et al., 1990] Ladkin, P., Anger F., Rodriguez R. Temporal reasoning with intervals in branching time. TR-90-028, International Computer Science Institute, 1990.

[Rybina et al., 2014a] G.V. Rybina, A.V. Mozgachev, D.V. Demidov Synergy of All-Purpose Static Solver and Temporal Reasoning Tools in Dynamic Integrated Expert Systems // Информационно-измерительные и управляющие системы, 2014. №8, С.27-33.

[Rybina et al., 2014b] Rybina G.V., Rybin V.M., Parondzhanov S.S., SoeThi Ha Aung. Some aspects of simulation application in dynamic integrated expert systems // Life Science Journal. 2014. V.11. №8s. P.145-149.

[Rybina et al., 2015] G.Rybina, V.Rybin Simulation modeling in the construction of dynamic integrated expert systems // International Journal "Information Models and Analyses" Volume 4, Number 1, 2015. P.30-40

[Shoham et al., 1988] Shoham Y. Goyal N. Representing time and action in AI. revised version of: Problems in formal temporal reasoning // Artificial Intelligence. 1988. N. 36(1). P.49-61.

[Shoham, 1987] Shoham Y. Temporal logics in AI: Semantical and ontological considerations // Artificial Intelligence. 1987. N. 33(1). P.37-63.

[Spranger, 2002] Spranger S. Representation of Temporal Knowledge for Web-based Applications. – Munchen, 2002.

[Емельянов и др., 1998] Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО. – М.: АНВИК, 1998. – 533с.

[Еремеев и др., 2004] Еремеев А.П., Троицкий В.В. Концепции и модели представления времени и их применение в интеллектуальных системах // Новости искусственного интеллекта. 2004. №1. С. 6–29.

[Еремеев и др., 2009] Еремеев А.П., Куриленко И.Е. Компонента временных рассуждений для интеллектуальных систем поддержки принятия решения реального времени // Искусственный интеллект и принятие решений. 2009. №1. С. 31-45.

[Еремеев и др., 2012] Еремеев А.П., Королев Ю.И. Сети Петри как инструмент для разработки интеллектуальных систем поддержки принятия решений реального времени // Гибридные и синергетические интеллектуальные системы: теория и практика. Материалы 1-го международного симпозиума - Калининград: Изд-во БФУ им. И. Канта, 2012, ч. 2, С. 108-117.

[Еремеев, 2006] Еремеев А.П. Логика ветвящегося времени и ее применение в интеллектуальных системах поддержки принятия решений. // Десятая национальная конференция по искусственному интеллекту с международным участием КИИ-2006: Труды конференции в 3-х томах.– М.: Физматлит, 2006. Т.3. С. 746-754.

[Осипов, 2008] Осипов Г.С. Динамические интеллектуальные системы. // Искусственный интеллект и принятие решений. 2008. №1. С. 47-54

[Осипов, 2011] Осипов Г.С. Методы искусственного интеллекта. М.: ФИЗМАТЛИТ, 2011.

[Плесневич, 1999] Плесневич Г.С. Метод аналитических таблиц для логики событий. // Труды Международной конференции "Интеллектуальное управление: новые интеллектуальные технологии в задачах управления (ICIT'99)".– Переславль-Залесский: ИПС, 1999.

[Рыбина и др., 2013] Рыбина Г.В., Мозгачев А.В., Паронджанов С.С., Со Тиха Аунг Динамические интегрированные экспертные системы: представление и обработка темпоральных знаний // Приборы и системы. Управление, контроль, диагностика. 2013. №6. С. 23-33.

[Рыбина и др., 2014] Рыбина Г.В., Рыбин В.М., Со Ти Ха Аунг. Некоторые аспекты применения имитационного моделирования в динамических интегрированных экспертных системах // Приборы и системы. Управление, контроль, диагностика. 2014. № 3. С. 35-45.

[Рыбина и др., 2015] Рыбина Г.В., Мозгачев А.В. Реализация темпорального вывода в динамических интегрированных экспертных системах // Искусственный интеллект и принятие решений. 2015. №1. С. 95-106

[Рыбина, 2008] Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. Монография. – М.: Научтехлитиздат, 2008. ‒ 482с.

[Рыбина, 2014] Рыбина Г.В. Интеллектуальные системы: от А до Я. Серия монографий в 3 кн.: Книга 1: Системы основанные на знаниях. Интегрированные экспертные системы. - М.: Научтехлитиздат, 2014. - 224 с.

[Рыбина, 2015] Рыбина Г.В. Интеллектуальные системы от А до Я. Серия монографий в 3х кн. Книга 2. Интеллектуальные диалоговые системы. Динамические интеллектуальные системы. – М.: Научтехлитиздат, 2015. – 164 с.

Authors' Information

Galina Rybina *– Doctor of Technical Science, Professor cybernetics department of National Research Nuclear University MEPhI (Moscow Engineering Physics Institute). RF President education award winner. Full member Academy of Natural Sciences. Accents on intelligent systems and technologies, static, dynamic and integrated expert systems, intelligent dialogue systems, multi-agents systems, workbenches; e-mail: galina@ailab.mephi.ru*

Victor Rybin *- Doctor of Technical Science, Professor department of Automation of National Research Nuclear University MEPhI (Moscow Engineering Physics Institute). Full member Academy of Natural Sciences. Accents on automation and electronics, electro physical complex, automatic control system, intelligent control systems, dynamic intelligent systems; e-mail: vmrybin@yandex.ru*