USING THE SIMULATION MODELING METHODS FOR THE DESIGNING REAL-TIME INTEGRATED EXPERT SYSTEMS

G. Rybina, V. Rybin

Abstract. *Certain theoretical and methodological problems of designing real-time dynamical expert systems, which belong to the class of the most complex integrated expert systems, are discussed. Primary attention is given to the problems of designing subsystems for modeling the external environment in the case where the environment is represented by complex engineering systems. A specific approach to designing simulation models for complex engineering systems is proposed and examples of the application of this approach based on the G2 (Gensym Corp.) tool system are described.*

Keywords: *integrated expert systems, real-time, simulation modeling, object-oriented model, rule, complex engineering systems, electrophysical complex.*

Introduction

The more and more sophisticated character of modem software systems is due to the fact that their architecture comprises a great number of subsystems and components with different functional characteristics, which interact with different groups of users. Because many components are created and developed autonomously, without any provision made for the possibility of joint operation and support of integration processes in the course of evolution of systems, this results, as a rule, in a substantial deterioration in the reliability of such systems. On the other hand, tendencies towards the integration of investigations in different fields, which have prevailed over the last ten years, have necessitated the integration of semantically dissimilar objects, models, methods, concepts, and technologies. This circumstance has inevitably led to the emergence of new classes of systems, such as integrated intelligent systems, integrated expert systems, integrated information systems, integrated manufacturing systems, etc.

Thus, research and development in the field of advanced integrated systems are of particular importance at present. Among the studies in this field, we can mention certain results on the theory and technique of designing integrated expert systems (IES) for static application domains (AD) obtained by the author in the course of conducting the AT-TECHNOLOGY research project (see, e.g. [Rybina,1997]).

However, the integration problems are most conspicuous in the construction of dynamic IES operating in a real-time mode (real-time integrated expert systems (RTIES)), because in this case it is necessary to ensure the following (see [Rybina,1998]): simulation of the external world and its various states; representation, storage, and analysis of time-varying data incoming from external sources; simultaneous temporal reasoning about several distinct asynchronous processes (tasks), support in functioning the inference mechanism under conditions of resource (time and memory) limitations, and other capabilities.

In this connection, special software tools (ST) are needed. These tools must make it possible to design and develop RTIES that can operate in dynamic AD, including the case where the correction of search strategies and knowledge acquisition are possible directly in the process of searching for a solution. The most widely known ST of such a kind are G2 (Gensym Corp.) and RT works (Talarian Corp.).

Over recent years, the author has accumulated certain experience in designing RTIES on the basis of the G2 tools for diagnostics and control problems, such as the control of modem electrophysical complexes [Rybin,Rybina,1998a, Rybin,Rybina,1998b], the diagnostics of complex engineering systems [Rybina,1998], the launch readiness verification of carrier rockets (prelaunch monitoring of carrier rockets) [Rybin,Rybina,1999], radioecological monitoring of areas adjacent to nuclear power plants [Kosterev et al, 1998].

By and large, despite the external dissimilarity of AD, complex engineering systems (CES) were studied. These systems are objects of a technical nature characterized by the following [Rybin,Rybina,1998]: their parameters constantly vary (in real time); they comprise from several hundred to several thousand functionally and structurally interrelated components, subsystems, modules, units, etc.; the diagnostics of these objects can be considered as a specific control process with the goal of determining the technological state of objects at each current instant (the general task of diagnostics of the object status) and, in addition, the task of fault finding (as a special case of the general diagnostic task); the functioning of these objects is a complex technological process accompanied by a multitude of abnormal conditions, rapid changes in the environment, and the lack of time for decision-making in response to abnormal conditions; a high price is paid for errors made by operators.

Therefore, any RTIES for diagnostics and control of CES (which are discrete and discrete-continuous for the main part) must ensure, in the general case, support for the execution of the following tasks: the dynamic modeling of all processes of functioning the of CES; monitoring the CES operation, detection of deviations from the prescribed regime, prefailure alerting and abnormal condition warning, emergency cut-out, etc.; studying the actions of the operators who control CES and training of personnel; a convenient graphic user interface for monitoring variations in the basic parameters characterizing CES operation, etc.

The architecture of an IES that is designed for real-time operation undergoes substantial changes due to these circumstances, because practically all basic components of a static IES are modified and two new subsystems are added—one of them is intended for the environment simulation, and the other supports the interface with the physical equipment.

The primary emphasis in the present paper is made on the problems of designing one of the most important components of the RTIES, the subsystem for simulation of the environment, which is represented by CES.

1. Statement of the Problem

The problems of real-time modeling of the external environment and its various states are very important in designing any RTIES. The methods and tools of simulation modeling (SM) are the most appropriate here, because the dimensionality of problems being solved and the unformalizability of complex objects and systems for which the RTIES are designed do not allow one to use rigorous mathematical methods.

On the whole, the SM is quite efficient, but, at the same time, it is a rather labour-consuming method and entails a number of problems such as the necessity to provide an adequate description of systems and processes in these systems, a correct interpretation of the results obtained, questions of stochastic convergence of simulation processes, the necessity to overcome difficulties caused by the problem of dimensionality, etc.

As was noted above, the application of methods and tools of SM in RTIES is caused primarily by the necessity for real-time modeling of the external environment and, in particular, to include the corresponding subsystems that adequately reflect all the processes and laws of functioning of the CES in the architecture of RTIES.

In [Rybina,1997] the author consider and solve these problems within the framework of the general-purpose problem-oriented methodology (POM). This methodology is intended for use in static and dynamic applications; it is a set of models, methods, algorithms, and procedures for designing of applied IES.

Since IES are implemented by integrating the methods and technologies of ES with those of conventional programming, the basic problem of integration within the framework of POM should be considered in the following way (see [Rybina,1997]): integration in the framework of IES of different components that execute formalized and unformalized tasks and determine the specific character of functioning of the entire IES (the top integration level); integration (functional, structural, and conceptual) related to basic system designs and concepts of development and design of particular classes of IES and their components (the medium integration level); integration (informational, software, and hardware) related to the technologies, ST, and platforms used (the bottom integration level).

Analyzing the problem of the top-level integration problem, we proposed the classification of IES and introduced the concepts of IES with the superficial and deep integration of components. It was also shown that the methodology for developing simple ES can be used only for designing IES with superficial integration and is completely inapplicable for IES with deep integration. In this case, we propose to apply the approach that improves ES by incorporating functions of a certain component N (where N is a database management system, an application package, SM system, etc.) that are unconventional for such ES, which is an important conceptual basis of the POM.

Here, the problems of integrating ES with SM in the framework of RTIES with deep integration of components are of primary interest, because, in this case, it is necessary to ensure the following (see [Rybin,Rybina,1998a]): the conceptual uniformity of approaches, models, and methods being used; the combination of rigorous mathematical methods of search for solutions with unformalized heuristic methods based on expert knowledge; due regard for the time factor both in the construction of models of AD and in the search for solutions, and other capabilities. The problems of designing RTIES that integrate simulation, conventional ES, as well as other constituent components of IES, are not clearly understood yet; therefore, the present paper deals with these problems. The modular principle of designing RTIES on the basis of POM, as well as the similarity of certain concepts used in ES and SM, makes it possible to integrate these technologies.

Below, we describe the results of experimental approbation of the elaborated models and methods of POM that ensure the integration of ES with SM in designing prototypes of RTIES for problems of control and diagnostics of CES; moreover, we focus mainly on one of the most complicated problems in the construction of RTIES, namely, the control problem for CES.

2. Construction of Simulation Models of CES

As noted above, the complexity of CES under study does not allow one to apply rigorous mathematical (analytical and numerical) models when describing these systems. There is no way in which this problem can be solved except by constructing a simulation model (MSM); moreover, preference should be given to the application of methods of the intelligent SM. Now, we consider an example of designing the simulation model MSM for the life-support and survival system (LSS) of the electrophysical complex (EPC) [Rybin,Rybina,1998a].

It should be noted that, due to their complexity, modern EPC are designed to eliminate the effects of subsystems on each other to the greatest possible extent; hence, each particular subsystem can be considered as a CES, and EPC can be treated as the set of CES, i.e., CES = $(CES_1,..., CES_N\}$. Therefore, the control system (CS) for a given object is a hierarchical CS whose control object at the top hierarchy level is the control system of a particular subsystem of the LSS.The simulation model M^{SM} can be subdivided into the model of internal stochastic perturbations (M_{SD}) and the model of the control system (Mcs). To simplify the description of the structure of the simulation model M^{SM}, we assume that the LSS comprises only one subsystem, for instance, the subsystem of vacuumizing. In this case, the simulation model M^{SM} of the life-support and survival system of the EPC will be represented by the simulation model M^{SM} of the subsystem for vacuumizing.

Thus, from the standpoint of SM, the life-support and survival system of the EPC is a discrete-continuous system. In the general case, the set-theoretic model of this system has the following form:

$$M^{SM}{}_{LSS} = \{M_{CO}, M_{CONTR}, M_{SD}, V^X, V^U, V^E, V^Y, V^Z, S, F^{Y \to U}, F^{XEU \to YZ}\},$$

where M_{CO} is the model of a control object, M_{CONTR} is the model of a controller; M_{SD} is the model of internal stochastic perturbations; $V^X = \{v_i^x\}$, $i = (1, m)$, is the set of monitored uncontrolled inputs; $V^U = \{v_j^u\}$, $j = (1, s)$, is the set of monitored controlled inputs of the M_{CO}; $V^E = \{v_h^e\}$, $h = (1, k)$, is the set of stochastic perturbations, $V^Y=\{v_l^y\}$, $l = (1, r)$, is the set of output parameters of the Mco (this set is used in the controller); $V^Z = \{v_g^z\}$, $g = (1, q)$, is the set of output parameters of the Mco; $S = \{s_c\}$, c = (1, n) is the set of possible (admissible and abnormal, i.e., inadmissible) states; $F^{Y \to U}$ is the function generating the control vector $u(t_{i+1})$ on the basis of the incoming output vector $y(t_i)$; $F^{XEU \to YZ}$ is a function mapping the input of the CO into its output.

Let I denote the input of M_{CO} (V^x, V^u, V^e), and let 0 denote the output (V^y, V^z); then, we have $o(t) = F^{xeu \to yz}(i(\tau)$, $\forall \tau \in [v, t]$, $o^{(k)}(v)$, $k = \varnothing; (n - 1), t)$, where $[o^{(k)}(t)(t=v)= o^{(k)}(v)$, $k = \varnothing; (n - 1)$, are the initial conditions (IC). Thus, at any instant t, the output is a certain function of the input and the IC.

The output of M_{CO} has the dimension $(r + q) \leq n$; therefore, $O \subseteq S$, and each particular state of the CO is described by the set of selected (on the basis of different criteria) properties (characteristics), i.e.,

$$C^{\sim} = \{c_1^{\sim},...,c_\lambda^{\sim}\}$$

where $C^{\sim}$ are the valuated properties of the CO. Thus, the set $C^{\sim}$ can be used to describe the set of states $S \subseteq S^{ad} \subseteq S^{ab}$, where S^{ad} is the set of admissible states and S^{ab} is the set of abnormal states.

This formalized representation of the simulation model M^{SM} of LSS of the EPC describes the operation of the entire system, but it is still too abstract for further implementation. In the present paper, in order to make the obtained model more specific and universal, we have used the Rational Rose Real-Time 6.0 CASE tool and the UML with enhancements for the support of real-time system engineering (UML-RT) is used as a language of model designing; these enhancements include the structural elements of the UML-RT-like Capsule and behavioral elements of type of Protocol. The idea is to represent all units using the diagram of classes and to write the diagram of states and transitions for each class. For instance, all units of the equipment making up M_{CO} and M_{CQNTR} are represented in the form of capsules (abstract representations of real-world objects, i.e., the equipment of the LSS of the EPC) with a necessary set of ports (abstract representations of data-transfer channels of real-world objects) through which the messages from other elements (from capsules in abstract declarations, from EPC's equipment in the real world) are incoming.

Since $O \subseteq S$, the output (V^y, V^z) of the M_{CO} is completely described by the attributes of capsules that represent M_{CONTR}; the component (V^u) is described by the capsules of the component representing the generator of stochastic perturbations; and each arrow (V^e) is represented by a message (information) transfer channel, i.e., by a protocol.

For each capsule, a diagram of states and transitions is constructed; the sets of these diagrams for all capsules are defined (represented) by the functions $F^{y \to u}$ and $F^{xeu \to yz}$. Thus, the representation of the simulation model M^{SM} in terms of UML-RT has the following form:

$$M^{SM}_{LSS} = <C, P, S, T, E, R_C, R_p, A>,$$

where C is the set of capsules; P is the set of ports, S is the set of states of capsules; T is the set of transitions; E is the set of events initiating a transition to another state; R_C is the set of relations between capsules, R_p is the set of relations between protocols; A is the mechanism for event tracing and the initiation of transitions (actions) corresponding to an event.

The representation of the model M^{SM} obtained above allows one to pass to its implementation at the level of tools; moreover, no constraints are imposed on the choice of ST for this purpose. In the present paper. the G2 system is used as a toolkit for implementing the simulation model M^{SM} of the life-support and survival system of the EPC and as a software development environment of the entire RTIES. In the G2 object-oriented environment, the above model is unessentially modified, namely: the diagram of classes, which was developed in the Rational Rose RealTime, is turned into an analogous diagram in the G2 environment; the hierarchy of protocols transforms into the hierarchy of connections and relations; the logic of the diagram of transitions and states is described using the G2 rules for the *whenever* construction; the application of these rules allows one to form the mechanism of *event tracing.*

An *event* means that the system is in one of the following a priori known states: a variable, a parameter, or an attribute of an object received a new value; an error occurred when a value was assigned to a variable; a variable lost its significance (the value is no longer significant); an object of some class was created; an object was moved (changed its coordinates) on the desktop; an object passed into an active or dormant state; two objects became related by a certain relation; two objects was connected to each other.

Therefore, M^{SM} in the G2 environment can be represented as the following set:

$$M^{SM}_{LSS} = <CL, O, C, E, RL_E, R_{C1}, R_0 >,$$

where *CL* is the set of system classes; *0* is the set of objects; C is the set of connections between objects; *E* is the set of model events; RL_E is the set of rules of the event-tracing machine; R_{C1} is the set of relations between classes; R_O is the set of relations between objects.

3. Methodology for Designing Simulation Models of CES

Thus, the following particular methodology for constructing the simulation model MSM of a CES, which is oriented towards use in the G2 environment, was developed within the framework of POM.

1. An AD is analyzed; the basic concepts of the AD, as well as the characteristics and operations of functioning of these concepts, are specified.
2. The abstractions of these concepts are described as classes in G2. The characteristics of concepts of a CES are represented by attributes, and the operations are represented by methods.
3. A powerful visual editor is employed to construct a scheme of the equipment of the real-world CES using the program objects and instances of described classes. Moreover, the interrelations of objects of the real-world CES are assigned by connections and relations.
4. As a result, one obtains the scheme $S = <0, R>$, where *0* is the set of objects of the scheme and *R* is the set of relations-between these objects.
5. Then, a set of model and temporal events, $E = \{e_i\}$ is constructed. A model event is an a priori specified state, i.e., the set of valuated attributes of one or several particular objects. A temporal event is a priori preset model or real time.
6. The set $D = \{d_j\}$ of actions, which are associated with the set of events, *E,* is constructed (an action is the totality of methods of objects of the scheme *S*), as well as the Scheduler (sequence monitor) of their joint functioning (sequential, concurrent, or with time delay). An action has a duration, which is realized in G2 through the use of the *wait for t* construction, where *t* is the delay time.
7. Based on Items 4 and 5, an event-tracing machine is constructed. This machine is applied for scanning the states of the system and for the initiation of an action associated with an event, when the system passes into a state for which this event is described in the set *E.* This mechanism is implemented in the G2 environment through the use of the rules of the *whenever* construction.
8. The set *B* of initial states is determined. The initialization of the scheme *S* with one of the initial states is carried out by executing constructions of the type *initially*.

4. Example of Application of Methodology for the Construction of Simulation Models of CES for RTIES

We use the example of modeling the LSS of the EPC for an operational prototype of the RTIES [Rybin, Rybina, 1999] to illustrate the elaborated methodology for designing simulation models of CES in the G2 environment. The LSS of the EPC, the charged-particle accelerator in the case under consideration, comprises interrelated subsystems for electric power supply, water supply, vacuumizing, magnet cooling, tunnel ventilation, high-frequency electric power supply, radiation protection, and fire safety.

We restrict ourselves to the consideration of the vacuumizing subsystem designed for the development and maintenance of vacuum in vacuum chambers. When the accelerator is started up, the air is evacuated in two stages; moreover, different types of pumps are used. At the first stage, a low vacuum is developed; at the second stage, the air is evacuated so as to develop a high vacuum. If an abnormal condition arises in the process of operation of the accelerator, i.e., if the pressure is at variance with that existing under high vacuum, then a specific control signal is sent to the emergency system and the vacuum chamber is blocked; the corresponding message is generated and sent to the control desk with the aid of the RTIES (see Fig. 1).

The construction of the simulation model M^{SM} of the vacuumizing subsystem is started simultaneously with the object-oriented analysis of the AD when the basic concepts of the AD are specified and the relations (their type, multiplicity, etc.) between objects are refined. In the case under consideration, one can distinguish the following: the vacuum chamber, the vacuum sector, and the exhaust units (VN1-MG, NEM-300, TMN-200). As has already been noted, the model is designed using, the Rational Rose for RealTime (although this is not obligatory and the designing can be carried out directly in the G2 environment). A diagram of classes is constructed; in our case, we apply the UML-RT, so this diagram is a diagram of capsules; moreover, the relations between capsules can be of the following four types: association, utilization, aggregation, inheritance.

Then, the interacting capsules and the message flows that describe their interaction are specified. For each such interaction, the concept of the message transmission channel and, as a consequence, that of the protocol of transmission of these messages are introduced abstractly. Thus, the diagram of capsules is refined by the diagram of protocols. Further, the sets of attributes that characterize the abstract state of a given capsule as a consequence of the state of the real-world equipment are specified, the conditions for transition from one state to another and the corresponding actions (i.e., the variation of a particular variable, the start-up of the capsule method, the transmission of a message to another capsule) are specified. On the basis of the information thus obtained, a diagram of transitions and states is constructed with the use of the UML-RT tools.

After this, the complete source information for the implementation of the obtained model in the G2 environment is available. In order to generate a list of events, it is sufficient to simply write out the conditions for transitions between states. The actions that are performed when a certain event occurs are also represented in the diagram of states and transitions using Rational Rose for RealTime. It only remains to connect the events and the actions associated with them by *whenever* constructions; these constructions represent one of the types of rules of the G2 system:

<*whenever rule*> :: = whenever <*event declaration*>
[or <*event declaration*>] [and when <*logical expression*>] then <*list of actions*>.

Then, a set of system states is constructed. To simplify the further description of the model, we assume that the vacuumizing subsystem can be in one of the nominal states only, for instance, the system is in the "off" state, which is described by the condition *none of the pumps is powered;* the system is at the stage of maintaining a low vacuum, which is described by the condition *the vacuum is in the range from* 10^{-3} *to* 10^{-2} *mm Hg;* the system is at the stage of maintaining a moderate vacuum, which is described by the condition *the vacuum is in the range from* 10^{-3} *to* $I0^{-6}$ *mm Hg;* the system is at the stage of maintaining a high vacuum, which is described by the condition *the vacuum is in the range from* 10^{-6} *to* 10^{-9} *mm Hg.*

Then, a set of actions is constructed in the G2 environment. The actions in this case are the G2 procedures that initiate and suspend the methods of objects and the methods themselves; for instance, the procedure that initiates all the methods of air evacuation used by pumps of the second type is declared as follows: *power_all_p2(VC),* etc.

And, finally, the event-tracing machine is constructed. In our case, the machine for event tracing and initiation works is as follows:

whenever the pressure of any vac_chamber VC receives a value and when the pressure of VC < 10 and the status of VC /= 2 *and the status of VC /= 3 and the status of VC /= 4 then conclude that the status of VC = 1;*

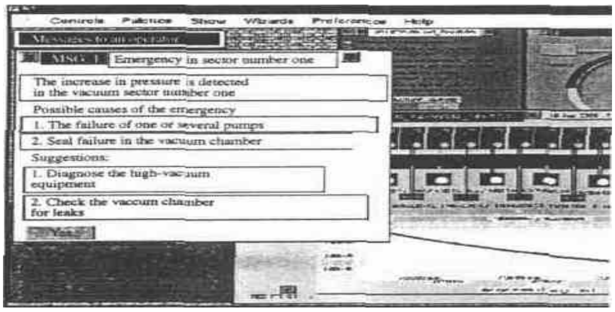
whenever the status of any vac_chamber VC receives a value and when the status of VC = 1 then conclude that the status of VC = 2 *and start power_all_p2(VC);*

whenever the pressure of any vac_chamber VC receives a value and when the pressure of VC < 10e-5 and the status of VC /=4 then conclude that the status of VC = 3;

whenever the status of any vac_chamber VC receives a value and when the status of VC = 3 then conclude that the status of VC = 4 and start power _all_p3(VC).

The initial conditions are determined using the rules of the *initially* type, which invoke the procedures of the system initialization when the applications are started up, i.e., *initially start initial_top_value (top_value) and start initial_bottom_yalue (bottom_value) and start initial_stub posts (stub_posts).*

An example of operation of the current version of the prototype of the RTIES for control of the EPC is presented in Fig. 1. This figure presents the case of detection of an abnormal condition in the operation of the vacuumizing subsystem. This condition is simulated with the aid of the simulation model M_{LSS}^{SM}; it is detected by the RTIES; and, using the rule-based inference, a message indicating possible causes of this condition and suggesting remedies for the trouble is issued to the operator.



Emergency in sector number one

The increase in pressure is detected in the vacuum sector number one

Possible causes of the emergency

1. The failure of one or several pumps

2. Seal failure in the vacuum chamber

Suggestions:

1. Diagnose the high-vacuum equipment

2. Check the vaccum chamber for leaks

Fig. 1. An example of operation of the prototype of the RTIES for the control of the EPC

Conclusion

Thus, a simulation model M^{SM} of any CES in the G2 environment is an object-oriented model; i.e., it is the set of program objects that simulate the dynamics of the behavior of the real-world CES described with the use of these objects; these latter are interrelated both by data transfer channels and by logical circuits. The real-time simulation process as such is supported by the G2 Scheduler (sequence monitor), which coordinates the processing of model and temporal events; this circumstance substantially facilitates the elaboration of the simulation model M^{SM}.

It should be noted that, if necessary, the MATLAB system or its analogs can be additionally used for modeling complex continuous processes described, in particular, by differential equations of the second or higher order; in this case, integration with G2 is carried out on the basis of tools of the GSI interface (G2 Standard Interface).

Bibliography

[Rybina,1997] G.V.Rybina. Task-Oriented Methodology for Computer-Aided Construction of Integrated Expert Systems for Static Problem Domains. Izv.Ross.Akad.Nauk, Teor.Sist.Upr. 5. 125-137. 1997.

[Rybina,1998] G.V.Rybina. Specific Features and Principles of Design of Integrated Expert Systems for Diagnosis of Complex Industrial Systems. Pribory i Sistemy Upravleniya. 9. 12-16. 1998.

[Rybin,Rybina,1998a] V.M.Rybin., G.V.Rybina. G2 Real-Time Expert System for Control of Electrophysical Complex. Proc. 2nd IMACS Int. Multiconf. CESA '98. Computational Engineering in Systems Applications, vol.1. Tunisia. 659-663. 1998.

[Rybin,Rybina,1998b] V.M.Rybin, G.V.Rybina. Using the Tools Complex G2 For Control of an Electrophysical Complex. Accelerator and Large Experimental Physics Control Systems. Proc. Int. Conf. on Accelerator and Large Experimental Physics Control Systems, Beijing (China), Zhaanel, J and Daneels, A., Eds.. Cern: Science Press. 107-109. 1998.

[Rybin,Rybina,1999] V.M.Rybin, G.V.Rybina. Dynamic Real-Time Integrated Expert Systems: Analysis of the Experience of 'Study and Development. Pribory Sistemy Upravleniya. 8. 4-8. 1999.

[Kosterev et al, 1998] V.V.Kosterev, E.A.Kramer-Ageev, G.V.Rybina, et al. A Prototype of Real-Time Expert Systems for Radio Ecological Monitoring of Areas Adjacent to Nuclear Power Stations. Proc. VI Nat. Conf. with Int. Participation II'98. Pushchino: RAII. V.2. 433-439. 1998.

Author information

Galina Rybina - Moscow Engineering Physics Institute (State University),Kashirskoe shosse, 31, 115409, Moscow, Russia,Email: galina@ailab.mephi.ru

Victor Rybin - Moscow Engineering Physics Institute (State University),Kashirskoe shosse, 31, 115409, Moscow, Russia,Email: rybin@aie.mephi.ru