A REAL-TIME DECISION SUPPORT SYSTEM PROTOTYPE FOR MANAGEMENT OF A POWER BLOCK

A.P. Eremeev, V.N. Vagin

Abstract: *This paper describes the basic tools for a real-time decision support system of a semiotic type on the example of the prototype for management and monitoring of a nuclear power block implemented on the basis of the tool complex G2+GDA using cognitive graphics and parallel processing. This work was supported by RFBR (project 02-07-90042).*

Introduction

Real time decision support systems (RTDSS) are hardware-software complexes, intended for the help to the decision making persons (DMP) at the management of complex objects and processes of a various nature in conditions of rigid temporary restrictions. When searching the decisions, expert models constructed on the basis of expert knowledge and heuristic methods of decision search are used. According to a modern classification of software, RTDSS are a class of integrated intelligent (expert) systems of a logic-linguistic type, combining strict mathematical decision search methods with non-strict, heuristic methods, based on expert knowledge [1-3].

The necessity of creating RTDSS is caused by continuously growing complexity of controlled objects and processes with simultaneous time reduction, yielded by DMP on problem situation analysis and acceptance of necessary managing actions.

Conceptually joining the approaches and methods of the decision support theory, theory of information systems, artificial intelligence and using the objective and subjective information, RTDSS provides DMP with the analysis of a soluble problem and directs him during decision search for increasing of decision efficiency.

One of the basic problems at designing RTDSS is a choice of the suitable formal apparatus for a description of the decision support process and a construction on its base adequate (correct) decision making model (DMM). As such apparatus the production systems are usually used. However, available expert system design tools are guided on static problem domains, i.e. on situations, not requiring corrections of DMM and decision support strategies during decision search.

The peculiarities of problems, solved by RTDSS, are:

- the necessity of the temporary factor account at the problem situation description and during decision search;
- the necessity of decision making in conditions of temporary restrictions determined by a real controlled process;
- the impossibility of obtaining all objective information, necessary for the decision, and in this connection use of subjective, expert information;
- the complexity of a search, the necessity of an active participation of DMP;
- the presence of nondeterminism, the necessity of a correction and an introduction of additional information during decision search.

A basic purpose of RTDSS is to help to DMP at the control of complex objects and processes, revealing and preventioning of dangers, development of the recommendations, i.e. to help in the sanction of problem situations before they will become irreversible.

The main design principles of RTDSS are:

- openness and dynamics;
- adaptivity and learning;
- semioticity;
- distributivity and parallelism in information processing;
- application of a highperformed computer technique and efficient tools (complete environments) of the type G2 + GDA (G2 Diagnostic Assistant);
- application of cognitive graphics and a hypertext in information mapping.

The generalized architecture of RTDSS is given in fig. 1. In contrast with traditional expert systems, in RTDSS it is necessary to include the additional modeling block, and the forecasting one for analysis, an estimation of accepted decision consequences and a choice of the best recommendations. These blocks are implemented on the basis of the imitative modeling system G2+GDA. The choice of the tool complex G2 for implementing RTDSS is caused by integration basic high-effective technologies of complex program product development: object-oriented programming; open system technology and client-server technology; the active object graphics; a structured natural language and a hypertext for the information representation; decision search, based on production rules, procedures, dynamic (imitative) models; parallel fulfillment in real time of independent processes; the friendly interface with various types of the users (DMP, system manager, expert, knowledge engineer, programmer); a combination of technology of intelligent (expert) systems based on knowledge, with the technology of traditional programming.

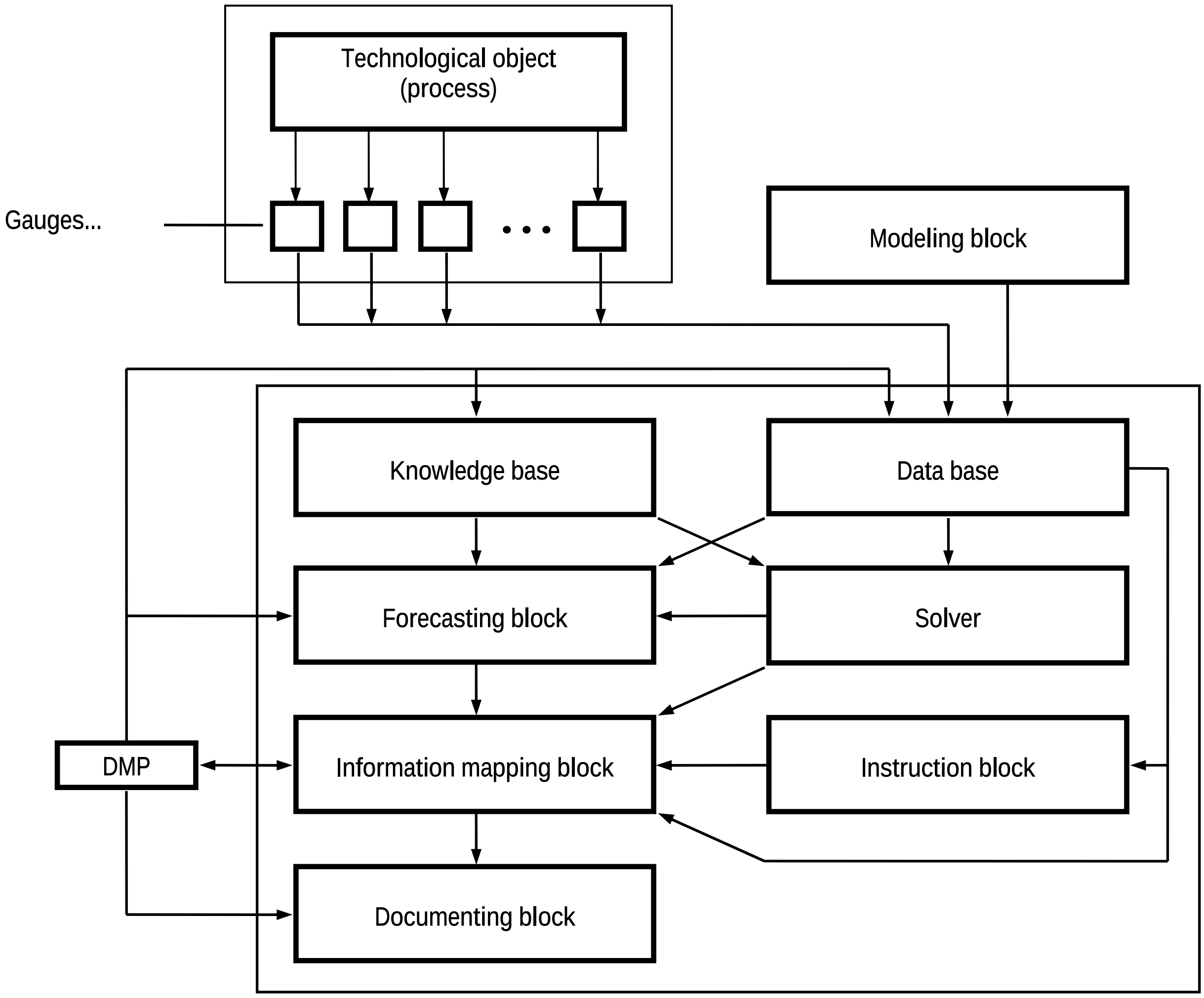


Fig. 1. The generalized architecture of RTDSS

The structure of base tools G2+GDA, necessary for RTDSS designing, consists of the interactive editor, tools of the graphic interface with an user, object-oriented graphics, graphic real time monitoring windows and animation, tools for display of connections between objects, interaction with an external environment, imitative modeling and processing of complex rules and procedures, tools for messages and explanations.

Such objects as a nuclear station power block are not made serially. Each object is unique and hence RTDSS for object management is also unique. But at designing RTDSS for various objects it is possible to use the same hardware platform and tools. Moreover, within the framework of G2+GDA class tools it is possible to design a tool environment of the same type of RTDSS. Such tool should give limited, but a rather complete set of primitives for the knowledge representation about an allocated class of objects and processes and about methods of management by them. Naturally the tool environment, oriented on dynamic RTDSS, should be open for updating by new constructive elements.

RTDSS of the semiotic type is defined by the collection

$$SS = \langle M, R(M), F(M), F(SS) \rangle, \text{ where}$$

- $M = \{M_1, ..., M_n\}$ is the set of formal or logic-linguistic models, implementing defined intelligent functions;
- $R(M)$ is the function for selection of the necessary model in a current situation;
- $F(M) = \{F(M_1), ..., F(M_n)\}$ is the set of modification functions of models $M_1, ..., M_n$;
- $F(SS)$ is the function for modification of SS system, i.e its base components M, R(M), F(M).

Applications of the first five design principles of RTDSS at implementing the prototype for monitoring and managing a nuclear station power block on the basis of tool complex G2+GDA was viewed in [4-10]. Here we present cognitive graphics and parallel processing means.

Cognitive Graphics in RTDSS

The instruction block (fig. 1) directs actions of DMP in planned transitive modes. It works automatically (on a situation) at switching on the appropriate mode. The information on a mode of object operation acts from a date base (DB).

The above requirements cause the necessity of the information representation in a knowledge base (KB) in the most convenient for DMP recognition a graphic form using the hypertext technology. The example of a fragment of external representation of KB for DMP as a decision tree with the necessary explanatory is submitted in fig.2, where 1, 2, ... 20 are block numbers in KB and selected items (1-2-3-6-13-19-18-20) map a control process. In a working mode activated by solver the chain of elements of a conclusion is allocated by other color. Between elements of KB in a working mode of the application, the relations can be established and be leafed.

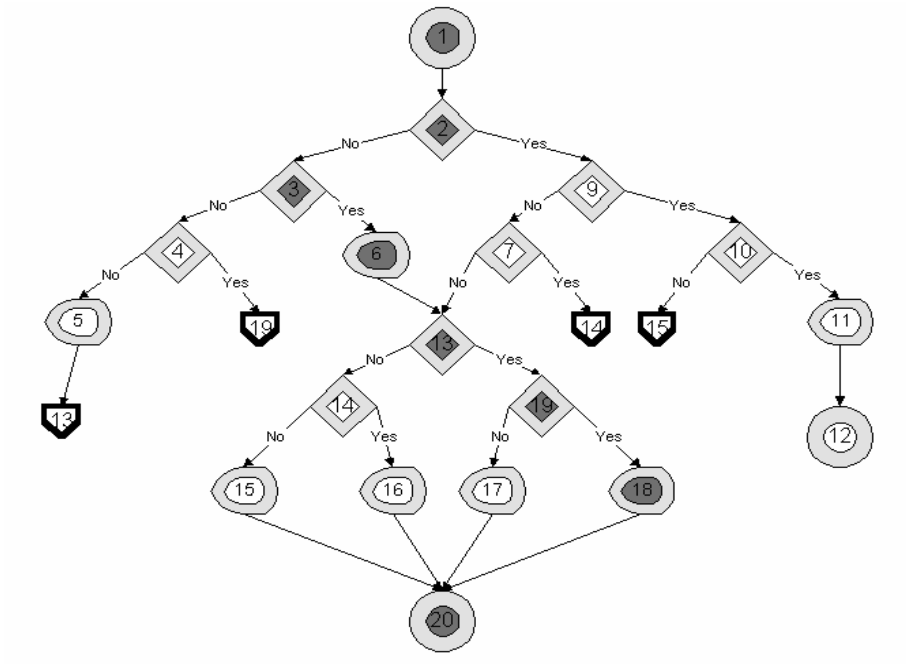


Fig.2. A fragment of the external representation of KB for DMP

The information mapping block carries out functions of information representation to DMP. Initial data for it are data from DB, results of an estimation of an object state, received by a solver, results of the forecasts, made by the forecasting block, and instructions given out by the instruction block.

The information, on the one hand, should be displayed in the form convenient for a fast recognition by DMP and, on the other hand, should be as it possible more complete. These requirements contradict each other, as the increasing of volume of the sign information decreases an ability of the person to perceive it. The problem is decided by means of the multilevel circuit of information mapping with application of cognitive graphics and a hypertext technology.

The information mapping unit for a prototype is the display of a workstation. The motionless image on the screen corresponds to a static object state, and the movement displays a transition of an object in a new condition, on that DMP owes immediately respond.

For mapping the information a number of working spaces is entered.

1. A working space with the scheme of subsystems of a power block (MCP – the main circulating pump, CAP – the capacitor, EJ – the ejector plant, SCRZ – the subsystem of cooling the reactor zone), concerns just to such class MCP (CAP, EJ, SCRZ), its auxiliary systems and gauges. A subsystem is presented by an icon of a class. The given class has a few ports for connection with elements of the pump auxiliary subsystems: an independent contour, circulation of oil in bearings, locking water, cooling liquids. The gauges are represented on groups by graphic images of parameters fixed by the subsystem. The working space of parameters becomes visible at pressing of a mouse key on a parameter icon. Dynamics of processes, occurring in a subsystem, is displayed by change of a graphic image color of parameters. Looking on the scheme, DMP can qualitatively estimate a state of a subsystem and define which parameters are outside of a range of allowable meanings (these parameters are allocated by the red color). For more detailed acquainting with a state of the subsystem, the operator has access to the parameter subworkspace.

2. A working space of the urgent messages. At a normal condition of the pump in a working space of the urgent messages there is only one message "System <name> is in norm". This message has a green background and does not signal about any anomaly. At occurrence of abnormal situations in a working space there are the appropriate messages on a red background.

3. A working space of KB. In it the decision tree is located. Basically this space is intended for an expert and a knowledge engineer for creation and testing KB.

In a decision making mode the means for a choice, concealment, moving and change of the volume of the specified working spaces are given to DMP. With the help of these means he can design project interface with the application.

In addition to DB, a message base is implemented which contains all diagnostic messages, that can be given to the operator in a decision making mode. The message base places in a separate working space and consists of copies of a class "subsystem-message", being a subclass of a built-in class G2 message with an attribute text. The class "subsystem-message" is complemented by an integer attribute "message-number" and a logic attribute "message actuality".

In the mode of decision making a DMP receives the means for choosing, hiding, transferring and changing a dimension of given working spaces. With the help of these means DMP can configure interface with the application.

Analysis of activity of operative-dispatching personnel has showed that it is preferably to use a three-level system of information representation on a control-labeled object or a process:

- a level of a system (or an object)on the whole, at which to be informed in what a (normal, abnormal or critical) state the system is and in what subsystems the deviations have arisen;
- a subsystem level at which a state of a particular subsystem appears;
- a level of directly measurable parameters with indicating not only parameter values but dynamics of their changing.

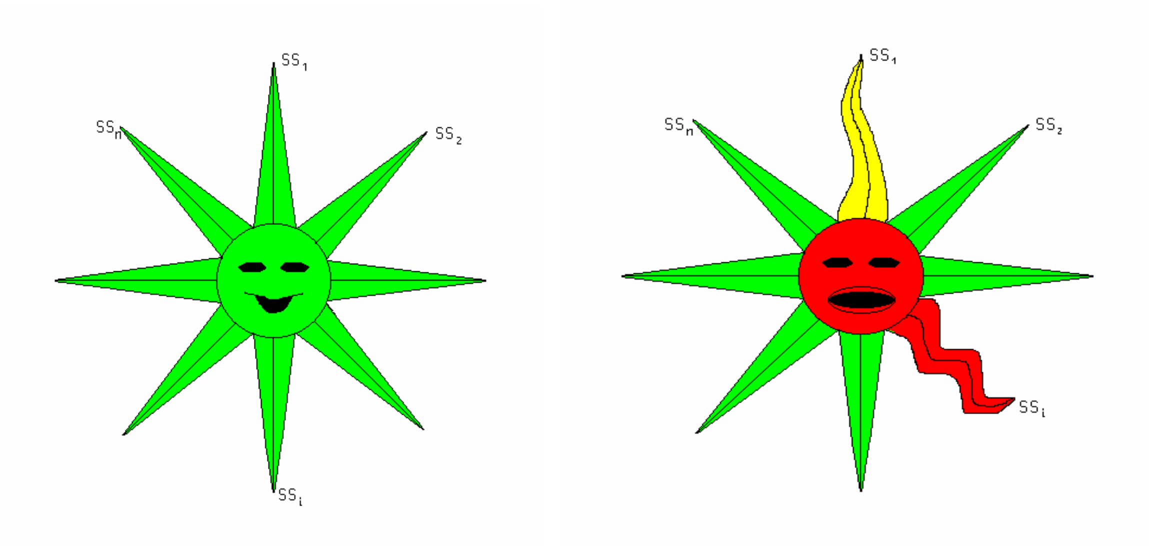
For building the graphic images a special editor is used. The basic type of image corresponds to a level of a system (an object) or a process on the whole and has a kind of the sun (kernel) with going out rays corresponding to subsystems (or generalized parameters of a process) SS_i of a top level. The number of rays is defined by a number of subsystems or generalized parameters and a thickness (a size of appropriate labels) of rays is determined by their importance.

The kernel may represent an image-face that can "smile", "wrinkle" or "cry" depending on an object state (normal, abnormal or critical accordingly) on the whole and its subsystems.

The kernel colour and rays may be red, yellow or green. The green colour of a kernel corresponds to a normal (on the staff) state (S_n) of the whole system (a process), the yellow one – to an abnormal state (S_{ab}) in presence of some deviations in subsystems and the red colour corresponds to a critical state (S_{cr}). In the state S_{cr} an immediate interference of DMP is necessary. If transferring in S_{ab} has occurred from S_n (i.e. system functioning has become worse), such interference is also necessary. The connection between states (colours) of subsystems and a kernel colour is given by means of production rules storing in KB. The examples of images of the first type corresponding to the whole system for normal and critical object states are presented in fig. 3a, b accordingly.

The image of the second type corresponds to a subsystem level and characterises its state. If a subsystem SS_i (a generalized parameter), in turn, is a complex system, then an image type coincides with the previous one and a process of "disclosure" may be recursively continued.

If the given subsystem SS_i is characterized by a collection of measurable parameters $\{P_j\}$, i = 1, 2, …, m, then the image of the third type occurs (see fig. 4) corresponding to the parameter level. The kernel and the external ring correspond to the zones of critical (upper and lower) parameter values and they are distinguished by red colour, the rings immediately adjoining to them present the zones of abnormal (upper and lower) parameter values and are marked by yellow colour and the central ring corresponding to the zone of



a – a normal state image b – a critical state image

Fig. 3. The image representation of a system state

normal parameter values has green colour. The rays present some normed indicators showing in what zone the values of appropriate parameters are found. The number and colour of zones are not strictly fixed and may be defined by DMP under adjustment.

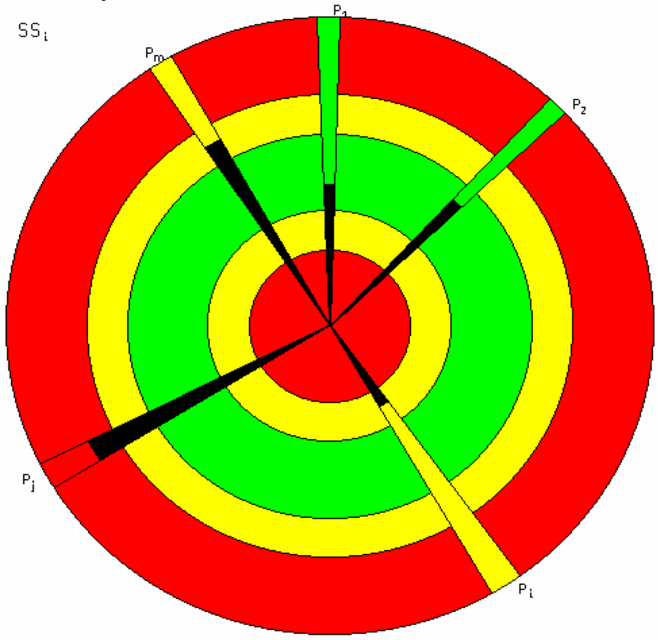


Fig. 4. The image of a parameter state

Besides the main image types there are means of switching on additional screens mapping dynamics of changing appropriate parameters in the form of graphics and provided by timers (fig.5).

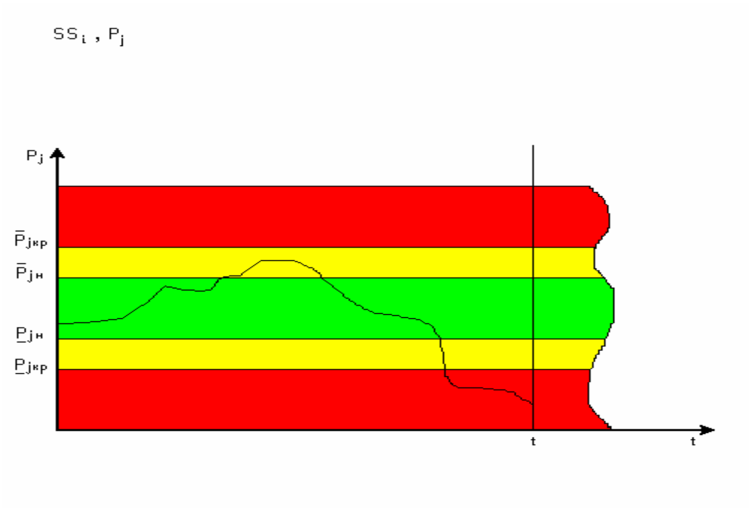


Fig. 5. The screen of changing a parameter

They allow receiving detail information about needed parameter values with the purpose of their normalization by means of some controlling actions. Note also, abnormal and critical states may be accompanied by appropriate sounds, twisting the rays changing the kernel "face" from "smiling" into "crying" and other additional means of paying attention to DMP.

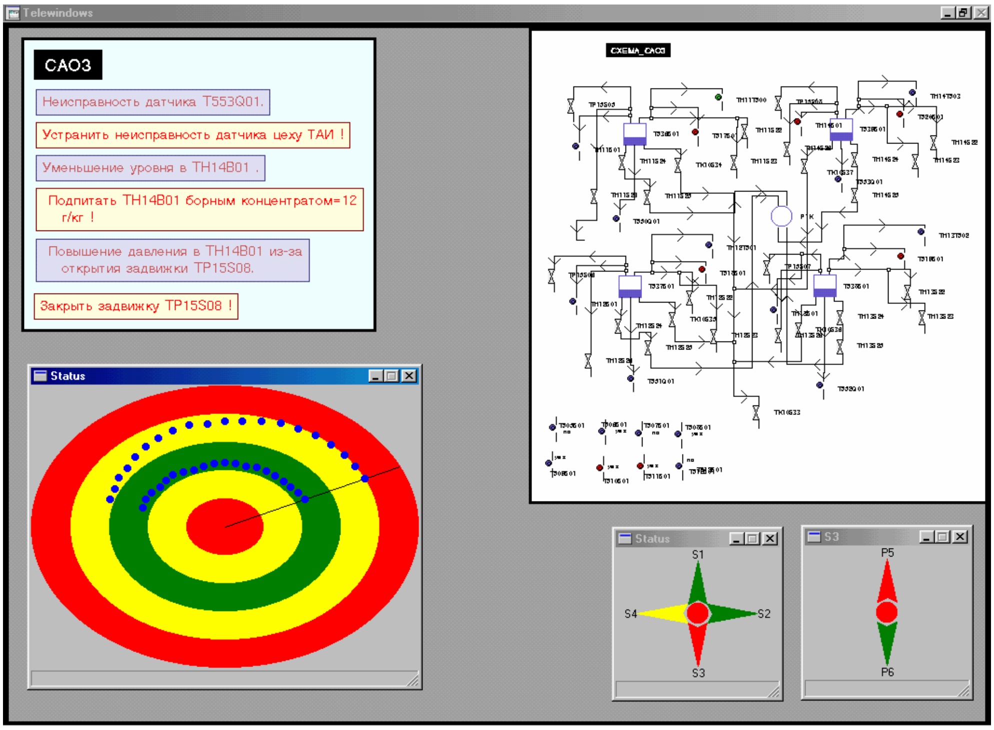


Fig. 6. The example of a poly-screen for DMP with cognitive graphics
(for the subsystem of cooling the reactor zone)

Setting up conformity between: a system state (an object, a process) on the whole and states of its subsystems SS_i, subsystem states SS_i and states of other subsystems, states of a subsystems SS_i and its measurable parameters P_j are accomplished by a special type of production rules containing in KB of the cognitive editor.

The example of a poly-screen for DMP with cognitive graphics is given in fig. 6.

Parallel Processing in RTDSS

At implementation of intelligent RTDSS the necessity of search of an acceptable decision in conditions of rigid temporary restrictions is displayed. It can be successfully implemented only by the use of hardware-software tools of parallel processing of information. We have focused on parallel inference methods and procedures because only parallelism of inference procedures allows to increase the speed of inference processes and to treat a huge amount of productions [10-12].

The prototype of RTDSS allows the parallel control for the systems – MCP, CAP, EJ, SCRZ. The following levels of external and internal parallelism are implemented.

Parallelism at the level of development. All systems of the nuclear power block realized in the prototype are being programmed and processed in parallel.

Parallelism at the level of problems, program modules and procedures. A few modules carrying out certain functions can work simultaneously within one application.

At parallel fulfillment of some problems one can arise the necessity of data exchange and accordingly a problem of data synchronization. For organization of synchronous interaction between blocks of various subsystems the production rules as **whenever** and **when** are used. For parallel processing inside procedures and methods the construction **do in parallel** is applied.

Parallelism at the level of rules. Opportunities of parallel processing of rules **initially**, **unconditionally**, **whenever**, **when**, **if**, **for** are used.

Parallelism inside rules. For parallel processing inside rules, i.e. for organization of fulfillment of certain actions for the whole set of objects of the same type, the generalized rules on the basis of constructions **for** and **every** are applied. Thus the same process is made active in parallel for the whole set of objects.

For organization of logic and arithmetic operations and their parallel processing (level of internal parallelism) blocks AND, OR, N TRUE (voting), synchronization and hold are used.

With the help of general constructions of inside rules and tools of parallel processing of the information in the prototype an opportunity of simultaneous browsing several screens on the polyscreen and simultaneous browsing the diagrams of parameter changes are implemented. Using the technology of polyscreens, DMP by reducing working spaces can browse the diagrams of all interesting parameters, an observable object and its systems.

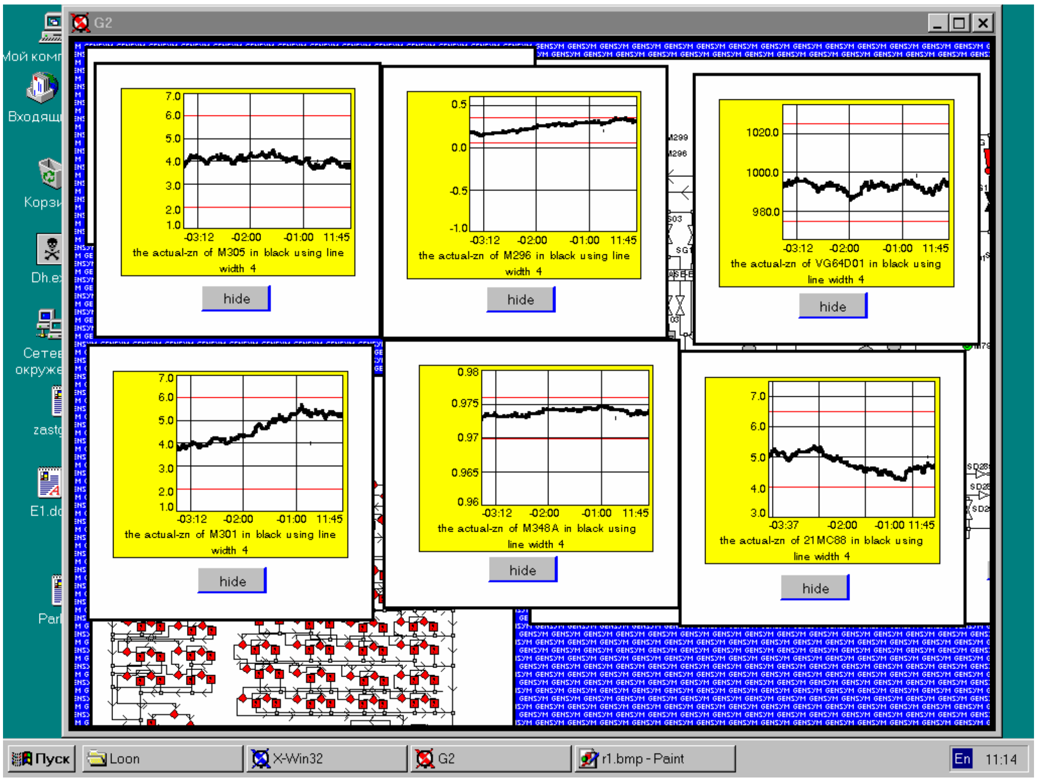


Fig. 7. The example of the polyscreen for display of parameters change dynamics

Sometimes the parallel processing is inadmissible, as the actions of rules can depend on each other and the fulfillment of one of them should precede fulfillment other. For example, it is necessary to change any parameter during imitation or forecasting of a real process, and then again to return in an initial situation. For such cases a construction **in order** is used, with the help of which the order of fulfillment of certain actions is established.

The external parallelism at the level of modules can be explained at the example of the module of parameter change forecasting (fig. 7).

For checking of a condition of a continuous parameter at its forecasting a simultaneous check is used, whether its urgent value above or below than established borders is. The parallel processing of input data allows to minimize the result calculation time.

Conclusion

During implementation of the RTDSS prototype for monitoring and management of a nuclear station power block, 31 classes of objects, 12 subclasses of variables, 2 subclass of connections between objects, 7 relations between objects were defined, 38 generalized rules, 45 procedures and 4 functions were written. For each continuous parameter, a subworkspace and an object for graphic display of dynamics of changing its values are defined. The cognitive graphics redactor was implemented as a separate module (block) and was connected with G2+GDA. The internal representation of the prototypes occupies 1 MB.

Bibliography

1. Vagin V.N., Eremeev A.P. Some Basic Principles of Design of Intelligent Systems for Supporting Real-Time Decision Making // J. of Computer and System Sciences International, Vol. 40, No. 6, 2001, pp. 953-961.
2. Vagin V.N., Yeremeyev A.P. Designing the Dynamic Decision Support Systems of a Semiotic Type // Proc. of the 1999 IEEE Internat. Symp. on Intelligent Control Systems / Intelligent Systems and Semiotics. Cambridge MA, Sept. 15-17, 1999, pp. 296-301.
3. Yeremeyev A.P. Organization of Semiotic Type Knowledge Representation Model for Dynamic Decision Support Systems // Proc. of Seventh Int. Conf. 'Artificial Intelligence and Information-Control Systems of Robots' AIICSR'97. Second Workshop on Applied Semiotics, Sept. 15, 1997, Smolenice Castle, Slovakia, pp 77-81.
4. Eremeev A.P., Shutova P.V. Learning and Adaptation in Real-Time Decision Support Systems of a Semiotic Type // Proc. of the IEEE Int. Conf. Artificial Intelligent Systems IEEE AIS'02, Sept. 5-10, 2002, Divnomorskoe, Russia, pp. 164-168.
5. Eremeev A.P., Vagin V.N. A Real Time Decision Support System for Monitoring and Management of a Complex Object Using Parallel Processing // Proc. of the IEEE Int. Conf. Artificial Intelligent Systems IEEE AIS'02, Sept. 5-10, 2002, Divnomorskoe, Russia, pp. 139-144.
6. Eremeev A.P., Tikhonov D.A., Shutova P.V. Support of Decision Making under Uncertainty on the Basis of the Non-Markov Model // Journal of Computer and Systems Sciences International, Vol. 38, No. 5, 1999, pp. 753-759.
7. Yeremeev A.P., Chibizova N.V. The prototype of a real time decision support system on the basis of the tool complex G2 // Proc. of Second Joint Conf. on Knowledge Based Software Engineering JCKBSE'96, Sazopol, Bulgaria, Sept. 21-22, 1996, pp.128-133.
8. Allachverdi N., Vagin V.N., Yeremeyev A.P. The Prototype of a Real Time Decision Support System for Monitoring and Management of a Nuclear Power Block // Proc. of 2nd Int. Conf. on Responsive Manufacturing ICRM-2002, June 26-28, 2002, University of Gaziantep, Turkey, pp. 602-608.
9. Vagin V.N., Yeremeyev A.P. Parallel Inference in Knowledge Representation Models // Proc. of Symp. on Roboyics and Cybernetics, CESA'96 IMACS Multiconference, July 9-12, 1996. Lille-France, pp. 184-188.
10. Eremeev A.P., Vagin V.N. A Real Time Decision Support System for Monitoring and Management of a Complex Object Using Parallel Processing // Proc. of the IEEE Int. Conf. Artificial Intelligent Systems IEEE AIS'02, Sept. 5-10, 2002, Divnomorskoe, Russia, pp. 139-144.
11. Vagin V.N. Parallel Inference in Situation Control Systems // Architectures for Semiotic Modelling and Situation Analysis in Large Complex Systems. Proc. of the 1995 ISIC Workshop. 10th IEEE Intern. Symp. On Intel, Control. Monterey, USA, 1995, pp. 109-116.
12. Vagin V.N., Yeremeyev A.P. Parallel Inference in Knowledge Representation Models // Proc. of Symp. on Roboyics and Cybernetics, CESA'96 IMACS Multiconference, July 9-12, 1996. Lille-France, pp. 184-188.

Author information

A. P. Eremeev - Applied Mathematics Department of the Moscow Power Engineering Institute (Technical University), eremeev@apmsun.mpei.ac.ru

V. N. Vagin - Applied Mathematics Department of the Moscow Power Engineering Institute (Technical University), vagin@apmsun.mpei.ac.ru