INT J COMPUT COMMUN, ISSN 1841-9836
8(1):166-175, February, 2013.

Issues on Applying Knowledge-Based Techniques in Real-Time Control Systems

D. Zmaranda, H. Silaghi, G. Gabor, C. Vancea

Doina Zmaranda, Helga Silaghi
Gianina Gabor, Codruta Vancea
University of Oradea
Romania, 410087 Oradea, 1 Universitatii St.
E-mail: zdoina@uoradea.ro, hsilaghi@uoradea.ro
gianina@uoradea.ro, cvancea@uoradea.ro

Abstract:
At the time being knowledge-based systems are used in almost all life aspects. The main reason for trying to use knowledge-based systems in real-time control is to reduce cognitive load on users (overload), their application proving to be important when conventional techniques have failed or are not sufficiently effective [1]. The development of automated diagnosis techniques and systems can help also to minimize downtime and maintain efficient output. This paper presents some issues of applying knowledge-based systems to real-time control systems. It describes and analyzes the main issues concerning the real-time domain and provides possible solutions, such as a set of requirements that a real-time knowledge-based system must satisfy. The paper proposes a possible architecture for applying knowledge-based techniques in real-time control systems. Finally, a way of employing knowledge-based techniques for extending the existing automatic control and monitoring system for the geothermal plant from the University of Oradea is presented.

Keywords: Real-time control systems, knowledge-based systems, Programmable Logic Controller

1 Introduction

Real-time systems generally consist of a series of complex, heterogeneous and critical processes. They are also closely coupled systems, consisting of a physical system part and a control computer system part. While the physical part reacts to the control signals from the computer, the control computer part and its software must interact with the dynamic properties of the physical part. Real-time systems are also, by definition, reactive systems. To increase their efficiency, different monitoring programs, tools, algorithms, and rules could be utilized. Generally, these programs are used for detecting abnormal behaviors, tracing workflow progress, and generating alerts and reports during the different phases of the system.

However, using knowledge-based techniques in real-time applications represents a major challenge because of several reasons [10]: problems related to time representation and reasoning about time; problems related to deadline because a knowledge-based system should provide the best solution within a given deadline; problems related to asynchronous evens handling that could lead to interruption of the inference process; problems related to integration of conventional real-time programming and knowledge-based programming.

Moreover, specific nature of real-time systems that implies interaction with external physical system that also implies specific features when including knowledge-based components [7]: the knowledge-based decision making system is hardly related to the external system; the knowledge-based decision making system should be also a real-time system in order to be sure that the decision making are performed before deadlines; implies both knowledge of control and of the external physical system each of them having a specific form; when dealing with complex real-time systems consisting of several sub-components, the problem of decision making need to be based on distributed knowledge.

Knowledge-based and real-time control technologies are complementary, rather than competitive technologies. Control technologies generally are oriented to quantitative processing while knowledge-based integrates both qualitative and quantitative processing [6]. Separating the description (the knowledge) of a process from the control algorithm allows knowledge to be more explicit, visible and analyzable, instead of being hidden inside of the procedural programming code.

Knowledge-based systems could be used for different purposes in real-time process control, the main domains of their applicability include [5] [9]: fault diagnosis, that implies detection, cause analysis and repetitive problem recognition; complex control schemes; process and control performance monitoring and statistical process control; real time Quality Management (QM); control system validation. For real-time control systems, an important issue is represented by their capability in fault detection and diagnosis because the availability and productivity can be significantly improved by shortening their downtime. Moreover, because personnels ability for observations could be incomplete or wrong and leading to incorrect diagnosis, intelligent system approaches need to be investigated and applied.

2 Knowledge-based techniques in real-time applications

Real-time applications could have different structures, and consequently, different approaches for using knowledge-based techniques can be employed. One typical structure of real-time control systems comes from the requirement of meeting the quest for automation and flexibility in complex manufacturing systems and is based on PLC (Programmable Logic Controller) usage. Such real-time control systems architectures are widespread because PLCs offer an adaptable and modular solution to the control problem. However, there are some shortcomings of this approach that are generated by PLCs inflexible programming system that do not support automatic analysis of logic circuits in order to look for a fault. Even if in todays modern PLCs there are some diagnosis functions available, their usage is limited and need to be extended. Consequently, developing a knowledge-based system for diagnosis purposes could represent a solution for implementing automated diagnosis techniques into complex manufacturing real-time systems. In order to create efficient solutions, several specific issues should be considered: knowledge representation and acquisition, real-time reasoning, knowledge validation, integration with real-time software [11] [12].

2.1 Knowledge representation and acquisition

The main reason for using real-time knowledge-based systems is to reduce cognitive load on users. Therefore, such systems require a knowledge representation that integrates several kind of knowledge taken from several sources: analytical models developed by using differential equations, material or energy balances or overall process behavior kinetics. Generally, each object could have a behavior that is represented by a combination of analytic (model-based) and heuristic (rule-based) statements. In PLC-based systems automatic monitoring elements available for faults are represented by discrete state signals into the PLC memory. These signals are indicating different operation states of the controlled plant and based on their values further diagnosis can be carried out. The values can be obtained by accessing PLC memory via a linkage from the computer that has implemented the diagnosis system. The diagnosis system must then use specific reasoning algorithms for searching all possible fault causes under the help of relevant knowledge and real-time data. Therefore, the knowledge acquisition task is very important, and could be made in two ways: artificial knowledge acquisition, model-based knowledge acquisition [3].

Artificial knowledge acquisition is obtained by knowing specific issues about the controlled plant. For example, in each plant there are several alarms that have the purpose of protecting plants equipments or preventing the plant to work in error conditions. These alarms are normally indicated by one or a combination of PLC signals. For example: temperature too high, pressure too low or a combination of these. Model-based knowledge acquisition is based on knowledge achieved during modeling the system behavior and constructing the PLC program, resulting in improvement of knowledge acquisition and diagnosis efficiency. Let define S the space that represents the fundamental set for all possible configurations of the control system variables. A specific configuration set s is given as:

$$s = s_1, s_2, ...s_n \in S \tag{1}$$

Let define the behavior space B that represents the fundamental set of all determinable behavioral attributes:

$$b = b_1, b_2, ...b_m \in B \tag{2}$$

Every subset F of B that comprise specific required profile for faulty behavior could be expressed by a combination of PLC signals associated with correspondent control system variables and their specific state. From formal point of view, these could be represented in the following way [2]:

$$fault- > (PLCsignal_1, state), (PLCsignal_2, state)...., (PLCsignal_n, state) \tag{3}$$

With every fault defined into the faulty behavior profile, all possible sources generating the fault are considered and specified. For example, if for a specific fault occurrence from the fault space F, device mapping dealing and inferring the functional relationship is considered:

$$F : D \rightarrow C_{relevant} \tag{4}$$

where D represents the device space and Crelevant represents relevant cause space.

The relationship between behavior space B, configuration space S and required profile for faulty behavior F is presented in Figure 1. Based on evaluation mapping between a specific behavior and profile of faulty behavior, if a match is found then the corresponding fault could be identified together with all possible causes.

2.2 Knowledge validation

In traditional real-time control systems, the control problem and its implementation through control algorithms is based on the exact knowledge of the control plant that is determined usually based on the mathematical model of the plant. Real-time knowledge-based control systems combine the analytical process model with conventional process control while reasoning about current, past and future situation in order to assess on-going developments and plan appropriate actions. Such systems allow the application to be structured into a model that is capable to behave and use its reasoning when taking a decision as human specialists do. Generally, a full control strategy requires not only variable (parameters) identification, state estimation and control, but also to check the validity of the data and process models before they are used in estimations. However, there is a relatively high degree of uncertainty concerning the plant starting from the mathematical model itself: there is not apriori knowledge of some parameters (for

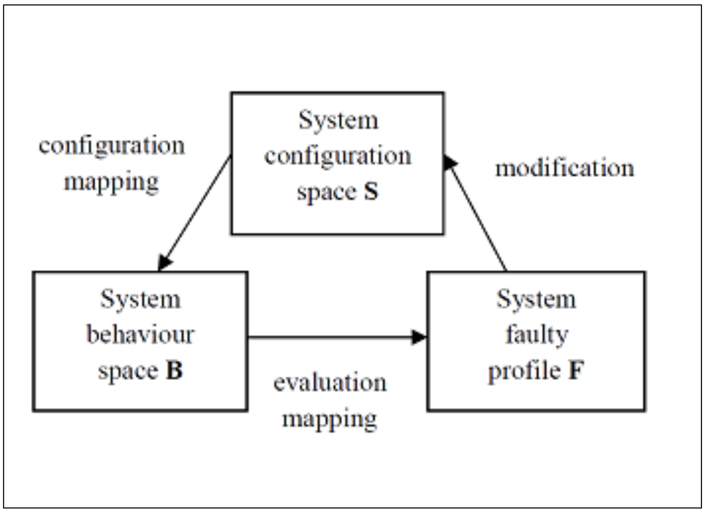
System configuration space **S**
configuration mapping
modification
System behaviour space **B**
System faulty profile **F**
evaluation mapping

Figure 1: The relationship between behavior space B, configuration space S and required profile for faulty behavior F

example, parameters for achieving stability conditions of a feedback control) or, the plant behavior may be not deterministic [4]. There is an important concern when using knowledge-based techniques for real-time control systems: the need for validating systems knowledge, allowing determination if it accurately represents an experts knowledge in the particular domain. In this idea, simulation, when available, could represent a very useful tool that provides a general overview of system dynamics.

2.3 Integrating knowledge-based software with real-time software

The real-time control software and rule-based software differ in their underlying execution models. Procedural software uses generally an imperative model in which software engineer determines the sequence of actions while rule-based systems represent a general control scheme of matching, selecting and execution of rules. Consequently, a knowledge-based real-time diagnosis system should be thought as an extension of existing control software that interacts both with knowledge database constructed by using artificial and model-based acquisition and with the real-time data acquired from PLC source code execution. A possible structure is presented in Figure 2.

Diagnostic reasoning is based on the knowledge base as well as real-time data from real-time database. The reasoning mechanism is based on logic control of faults. Thus, it uses the logical expression for faults presented in (3) where each term represents a possible cause of fault indicated by a specific PLC signal. By comparing the fault state with the current state of signals from real-time database in PLC, an occurrence of a fault state could be identified. Furthermore, by using systems profile mapping to devices and associated causes, as presented in (4), associated devices and causes of the fault could be shown.

2.4 Real-time reasoning

First attempt for using knowledge-based systems for real-time process control involves using static expert systems that take a snapshot of plant data. Static expert systems use pattern matching of a set of facts and rules. With no time constraints, this approach proves to be practical but when time constrains come into picture, this might not be a good idea. The

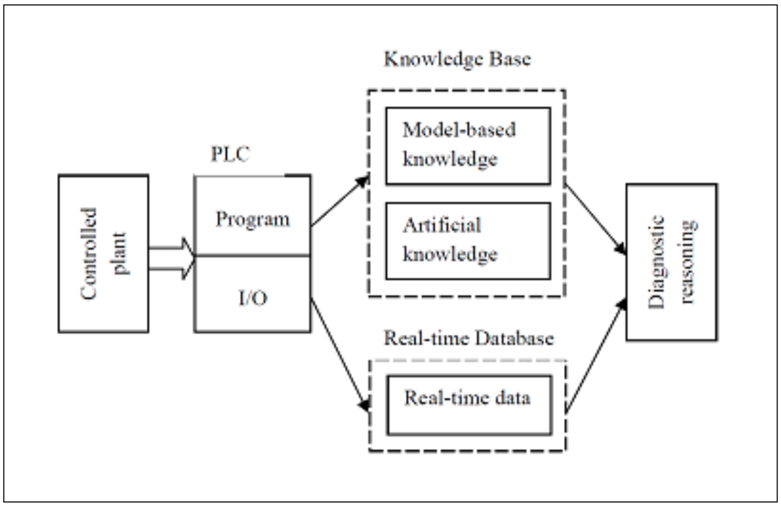

Figure 2: Structure of a diagnostic system for a real-time control system

elements that should be considered in this situation are: temporal reasoning and responding within a given response time.

In the controlled system, variables such as temperatures and pressures may vary in time; therefore, a time representation together with the possibility of reasoning about time is essential to be included. So, to the elementary entity of knowledge base additional time information should be attached. Also, the rules defined into the expert system may be extended with specific, temporal-extended rules, such as: operators *always*, to formulate the premise of a rule; qualitative statements that refer to the relation of time points or time intervals, without exact time specification: *earlier*, *after* and similar; quantitative statements that allow expressing conditions with specification of exact point of time: for example *at 12:00:00 p.m.*

Generally, the basic characteristic of a system that guarantees a certain response time is represented by its determinism. Knowledge-based systems are by their nature non-deterministic because the time of inference is dependent on the given situation. If imposed real-time deadlines are lower than the maximum searching time that is needed for a certain inference, the response time requirements cannot be met.

In order to meet those requirements, three strategies could be applied: implementing algorithms to quantitatively estimate the maximum searching time, reducing the inference searching time or define an embedded diagnosis system approach which will integrate the diagnosis models into the PLC control program so that faults could be diagnosed in real-time.

The problem of including all diagnosis into the PLC has also disadvantages by creating a much more complex control program. Integrating inference rules into the control program itself complicate the rules and makes introduction of new rules more difficult. In addition, the process of integration of different knowledge and information of these systems represent a tedious process. A mixed approach could be a possible better solution: only for critical faults, the diagnosis part with corrective actions should be included into the PLC and the rest of them should remain in charge of the knowledge-based system. In the next chapter, a way of employing knowledge-based techniques for extending the existing automatic control and monitoring system for the geothermal plant from the University of Oradea is presented.

3 The automatic control and monitoring system structure

The automatic control and monitoring system of the geothermal plant from the University of Oradea is an example of a real-time control system that has been developed using a combination between a PLC (Programmable Logic Controller) and a PC (for the user interface and supervisory control). From the structural point of view, the controlled plant is composed of 3 parts: the well station, the pump station and the heat station.

The system functions in the following way: first, the geothermal water is extracted from the well station using a deep well pump if the necessary flow rate is greater than the artesian one; the water is then stored into a reservoir tank, which acts as an accumulator and also separates the production network from the distribution network. Then, from the reservoir tank the water is pumped, through the pump station to the heat station and further, in the heat station, the water is not directly utilized, but through 4 heat exchangers; the water that comes out from these heat exchangers flows into the distribution network and heats the university campus buildings.

The structure of the control system consists of a PLC for controlling the geothermal heating system based on a control program embedded to the controller, connected with a PC computer and contains the user interface for the operator that is implemented using Wonderware InTouch software [15]. The InTouch display management subsystem handles display call up, real-time display update, data entry and process schematics. It also maintains the PLC real-time and historical database that could be used to follow the time evolution of certain parameters or for statistical calculations. The real-time database provided includes maintenance of historical data in addition to current values of process variables. In the current implementation approach, the decision process in order to achieve possible solutions for several faults, such as sensors and other equipments faults implies changing the plant operation into a safety mode and employing the operator for tracing the fault, its effects and related causes. Also, PLC control program deals with diagnosis to only of few critical faults. Based on current situation of the geothermal plant and the current research in the domain, a proposal for extending the existing control system with a knowledge-based system is developed and described further in this paper.

3.1 Analysis phase. Knowledge acquisition and validation

The main issue when constructing a knowledge-based system is the way on which description (knowledge) is built up in accordance with the plant behavior and structure. Consequently, the development of a knowledge-based system for the existing plant and control system implies selecting the most important characteristics of the system that will be used in order to construct the knowledge database. Moreover, there is the need for validating systems knowledge, that means determination if it accurately represents an experts knowledge in the particular domain. In this idea, the simulator developed for the geothermal plant of the University of Oradea proves to be of great help. The simulator, which was previously developed, provides a simplified physical model of the plant dynamics together with PLC control, which is formulated into an easier-to-operate computer simulation: the ACSL solution to the model equations. Key elements of the improved easy-of-operation are the use of general-purpose simulation language ACSL (Advanced Continuous Simulation Language), and pre-programmed modules of all important plant components, including control elements [16]. In the development phase of control system, the simulator has provided a useful tool for testing the system specifications, including the adopted control strategy; it could be employed also into the process of step-by-step knowledge acquisition and for further validation and updating. Based on information gathered from simulation and from the control program development, knowledge acquisition could be achieved. For example, if we refer to possible faults, a structured notation could be used when doing knowledge aquisition, based on tables, as shown in Table 1.

Table 1: Faults and associated switches in PLC (partial)

Digital inputs	Fault notation
Overload protection switch pump P3/P4	P3_OL, P4_OL
Thermal switch pump P3/P4	P3_TD, P4_TD
Overheat in the heat station	TS2
Local emergency stop switch in the pump station panel	ES1

3.2 Design phase. Knowledge representation

Knowledge acquisition is the most important part when developing a knowledge-based system, but an important point is represented also by the way on which knowledge representation is done. Representation is tied to the production rules that should be expressed accordingly to the knowledge system that is used. Current knowledge-based industrial systems are generally built within shells, which package a combination of tools. Different shells may include different features useful for real-time control applications, such as: hierarchy for objects, associative knowledge, relating objects in the form of connections and relations, rules and associated inference engine, analytic knowledge, such as functions, formulas, and differential equation simulation, real-time features such as time stamping and validity intervals for variables, history-keeping, run-time environment. In this idea, current literature presents several expert systems that could be used for real-time process control systems, the most known being G2 and JESS.

G2 real-time expert system [13] allows integration of models and rules for combining model-based and artificial knowledge representation and it is based on an inference engine that can use generic forms of knowledge, interpreted for specific instances in the domain. It is specifically designed for process control and related applications and allows the process engineer to implement and manage the expert system. But, even if G2 is claimed to be real-time, there is no mention of verifiability from temporal point of view.

Java Expert System Shell or JESS [13] is inspired by the artificial intelligence production rule language CLIPS being a fully developed Java API for creating rule-based expert systems. Even if it is architecturally inspired by CLIPS, it exhibits a LISP-like syntax. It consists of three components: the rules (knowledge base), the working memory (fact base corresponding to real-time base) and an inference engine (rule engine). JESS uses the Rete (ree-tee) algorithm to match patterns. Rete algorithm is not time predictable but two newer algorithms Treat and Leaps have introduced some optimization from this point of view; there are as well not time predictable but, with some restrictions, it would allow to be time predictable.

Consequently, we consider to develop a solution based on JESS for our system. Thus, a knowledge model can be constructed starting from observing the main problems associated with every specific item (for example, pumps P3/P4 from the pump station) in the operational system and creating the model (Table 2) that presents correlation between problems and items that can be solved by expert system.

Table 2: Correlation model

SLOT	ITEM	PROBLEMS
Situation (for pumps P3/P4)	Engine	Overheating, Vibration, Lack of voltage

Afterwards, for each identified problem, the possible causes are identified and associated, generating a knowledge model (Table 3).

Table 3: Knowledge model

SLOT	INFERENCE LEVEL	TASK LEVEL
Cause	Overheating	Cooler water missing, Too big temperature in pump station
	Vibration	Bend axle, Motor basement snap
	Lack of voltage	Disruption of wires, General pump station switch off

After establishing the knowledge model, the design phase is dedicated for structuring the rules according to requirements of JESS inference engine. JESS uses the notion of frames that are hierarchical representation that includes several components (slots, facet, datum and how). Frame definition is made by using Deftemplate, having the following generic form [2]:

```
((deftemplate <deftemplate-name> [extends <classname>]
[<doc-comment>]
[(slot <slot-name> [(default | default-dynamic<value>)]
[(type <typespec>))]*)
```

For example, the template for pumps P3/P4 will look in the following way:

```
(deftemplate P3/P4
(slot situation (default NF))
(slot cause (default NF)
)
```

, where NF states from Not Found. JESS basis of knowledge composed from rules could be further built based on the previously frame definitions. These slots in the JESS rules structure will be unified by using pattern-matching by the inference engine through the Rete algorithm [8].

3.3 Overall proposed architecture

The proposed modifications of the existing architecture are illustrated in Figure 3. It employs as additional element an additional computer on which the JESS Expert System is running that is connected with the PC on which the user interface was developed and on which PLC historical and real-time database resides. Historical and real-time databases are, in our situation, created and updated by InTouch in real-time, based on PLC associated inputs.

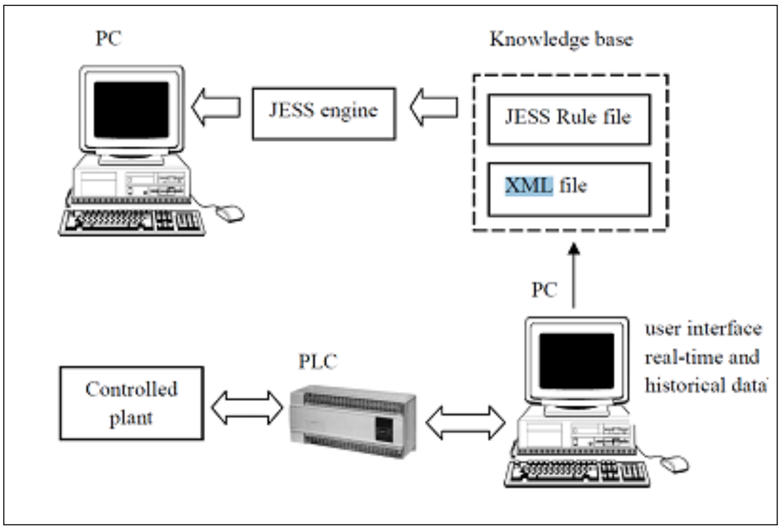

Figure 3: The Knowledge System architecture

In the proposed architecture, the main issue is represented by the way in which collaboration of several components is achieved, because PLC real-time database has a specific format for storing, defined by InTouch. There is a need of conversion from this format in order to be able to interact with the knowledge base that resides on the expert systems PC computer. A possibility is represented by using an XML file that stores all the tagnames of the faults, generated from PLC database. The idea is to use the XML file to feed the knowledge base and to interact as an intermediate level between expert system and PLC real-time database. Then, this file could be used as input by a JESS function in order to check is value; afterwards, for all tagnames that are acted (indicating a fault) rules pattern should be evaluated. Consequently, the architecture colaboration of components could be achieved by integration of JESS engine with the real-time database through the XML file.

There are several issues that are not completely defined in our architecture. For example, the way in which the XML file is generated based on PLC real-time database in still in question. Also, after completing the implementation, system performance should be evaluated in order to prove that the solution provides satisfactory.

4 Conclusions and Future Works

Knowledge-based systems are making significant contributions to real-time process control applications. Their applications are often in areas which complement traditional process control technology, like, for instance, diagnosis and handling abnormal situations. They integrate knowledge-based techniques with conventional control, having significant benefits in overall quality management. But, a knowledge-based system operating in a real-time situation will typically need to respond to a changing environment involving asynchronous flow of events and dynamically changing requirements with limitations on time, hardware, and other resources. Determining how fast this system can respond under all possible situations is a difficult problem that requires using flexible software architecture in order to provide the necessary reasoning on rapidly changing data. In this paper, various issues on applying knowledge-based techniques in real-time control systems have been presented; starting from this foundation, an implementation structure of a knowledge-based system for the existing automatic control and monitoring system for the geothermal plant from the University of Oradea was analyzed and an overall architecture is proposed for further implementation. The actually proposed knowledge-based system approach is focused mainly on general architecture and component colaboration. The architecture does not include time-constrains validation, it relies only on the performance of the JESS engine; a further solution for creating a specific expert system that includes temporal reasoning will be also investigated in the future.

Bibliography

[1] Singh A., Verma M., Real Time Expert System - Its Applications, IJCST, ISSN : 2229-4333 (Print), ISSN : 0976-8491 (Online), 1(2):150-153, 2010.

[2] Farias O., Labidi S., Neto J. F., Albuquerque S., *A Real Time Expert System For Decision Making in Rotary Railcar Dumpers*, Automation Control - Theory and Practice, A D Rodi (Ed.), ISBN: 978-953-307-039-1, InTech, available from: http://www.intechopen.com/books/automation-control-theory-and-practice/a-real-time-expert-system-for-decision-making-in-rotary-railcar-dumpers, 2009.

[3] El-Desouky A. I., Arafat H. A., Laban S., *Implementing Knowledge-Based Systems in Monitoring Real-Time Systems*, Geophysical Research Abstracts, Vol. 10, EGU2008-A-12044, SRef-ID: 1607-7962/gra/EGU2008-A-12044, EGU General Assembly, 2008.

[4] Eremeev A., Varshavskiy P., Case-Based Reasoning Method for Real-Time Expert Diagnostics Systems, *International Journal Information Theories and Applications*, 15:119-125, 2008.

[5] Wai K. S., Latif Abd., Rahman B. A., Zaiyadi M. F., Aziz A. Abd., Expert System in Real World Applications, http://www.generation5.org/content/2005/ExpertSystem.asp, 2005.

[6] Bubnicki Z., *Modern Control Theory*, ISBN 10 30540-23951-0, ISBN 13 978-3-540-23951-2, Springer-Verlag, Berlin, Hedelberg, 2005.

[7] Bubnicki Z., *Kowledge-based and learning Control systems*, Control Systems, Robotics and Automation, Ecyclopedia of Life-Support Systems, vol. XVII, 2005.

[8] Grissa-Touzi A., Ounally H., Boulila A., *VISUAL JESS: AN Expandable Visual Generator of Oriented Object Expert systems*, World Academy of Science, Engineering and Technology, 2005.

[9] Alexander A. J., How to Use Expert Systems in Real World Applications, http://conferences.embarcadero.com/article/32093, 2004.

[10] Vagin V.N., Eremeev A.P., Certain Basic Principles of Designing Real-Time Intelligent Decision Systems, *Journal of Computer and Systems Sciences International*, 40(6):953-961, 2001.

[11] Laffey T. J., Cox P. A., Schmidt J. L., Kao S. M., Read J. Y., Real-Time Knowledge Based Systems, *AI Magazine*, 9(1):27-45, 1988.

[12] Charpillet F., Look on the Issue of Building Real-Time Knowledge Based Systems - Research Summary, Association for the Advancement of Artificial Intelligence Technical Report WS-97-06, 1997.

[13] http://www.gensym.com/

[14] http://herzberg.ca.sandia.gov/jess/

[15] http://global.wonderware.com/EN/Pages/WonderwareInTouchHMI.aspx

[16] MGA software, *Advanced continuous simulation language (ACSL) - reference manual*,1995