# Support System for Modeling and Design of Control Systems

**Baltazár Frankovič, Ivana Budinská, Jolana Sebestyénová, Viktor Oravec**

Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
e-mail: frankovic@savba.sk, budinska@savba.sk, sebestyenova@savba.sk, upsyviki@savba.sk

*Abstract: In the paper a decision support system is described which consist of users questionaire (it containing the user's requirements and the main characteristics of the controlled process). The core of the systems is created by a database of modeling, control and simulation tools (algorithms) for manufacturing processes and an intelligent decision system, generic block, which provides relevant reasoning about suitable algorithms and tools in respect of user's requirements and description of processes.*

*Keywords: case based reasonong, decision making, multi-agent system, database, ontology*

## 1 Introduction

The decision support system, MARABU, developed at the Institute of Informatics in cooperation with Slovak Technical University Bratislava and Technical University Košice, is a modular multi-agent system for modeling, control design and simulation of production systems. It is intended for experts and technical staff from manufacturing enterprises, as well as for students of technical schools; it might be used for solving critical situations in quasi real time, and/or for learning and training technical staff and students of technical universities. The design of a production system control includes three phases. First a model of production system has to be designed, then a control algorithm has to be suggested, and finally the proposed solution should be simulated and verified. This procedure determines the basic architecture of the MARABU. There are many problems related to the system development. For start, knowledge has to be represented in a form that is either human- or computer-readable. For Second, all data and knowledge about specific domains has to be stored, archived and organized for future re-use. Next, it is important to create new knowledge based on the stored and archived information. The scope of the paper is to present a multi-agent architecture to handle knowledge and provide decision support for modeling, control and simulation. The knowledge is organized in three levels:

- The process level includes: power engineering processes, chemical engineering processes, gas plant processes, manufacturing and transport processes, food technology processes;
- The process model level includes: continuous, discrete, hybrid, linear, event driven, nonlinear, deterministic and stochastic processes;
- The synthesis method level includes: event driven control, process control, supervisory control.

The system works in two modes:

1 finding an appropriate tool, method, or algorithm, searching in a case- base

2 connecting an external application.

Therefore, the database is organized as a structured database. The database for tools, algorithms and methods stores metadata for each item.

## 2 MARABU Architecture

The MARABU architecture is shown in Figure 1, and one possible implementation is shown in Figure 2, where the following blocks and agents are considered:

**Users agent (UA)** – provides an intelligent interface between the designer (user) and the other agents in the system. UA receives designer's requirements and description of the process to be controlled as inputs for reasoning and decision making, and returns suggested solution(s) for the user. The user interface is built on the basis of so questions and answers. The system of questions creates questionnaire. The **questionnaire (Q)** is built similarly to an expert system; the users follow the steps in the requirements by answering the questions. On the basis of the answers, the next level of the questionnaire is opened for him/her.

**Modeling Agent (MA)** – after receiving information about the process, the MA searches for appropriate modeling tools and algorithms on the basis of previous cases. It returns suggested algorithms and ask for more precise information according to a chosen algorithm. The final decision on which algorithm and/or tool has to be chosen is up to the user. Finally, the modeling agent returns a model of the described production process.

**Control Agent (CA)** – receives a finally chosen model with all necessary attributes defined. On the basis of the model, the CA searches for an appropriate control algorithms in the database. Through the CA, MARABU system negotiates with the user and finally chooses appropriate control algorithms. The user has to choose an algorithm from the suggested ones and specify all values needed for that. The CA returns the control algorithm for the process according to the user's specifications.

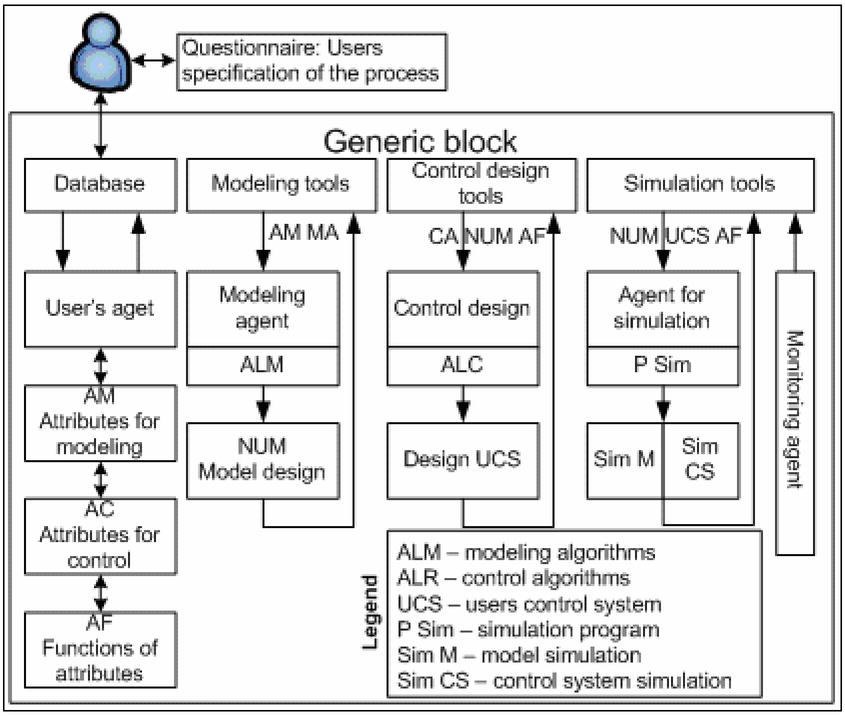


Figure 1
The MARABU system architecture

**Simulation Agent (SA)** – is responsible for simulation of control for the chosen model and control algorithm with the aim to help the designer assess the proposed solution.

**Monitoring Agent (MoA)** – follows the system behavior after applying the recommended method for designing the control. If all the requirements are satisfied, then the MoA updates the database by newly achieved results. It means the system stores all solutions for next re-use and application. Otherwise, the MA and CA have to repeat their calculations.

**The above described agents create so called Generic Block** – the core of the system (see Figure 1). A Generic Block (GB) is a multi agent system, which is responsible for managing the activities of all other agents, and provides interfaces among the system, users, and external applications.

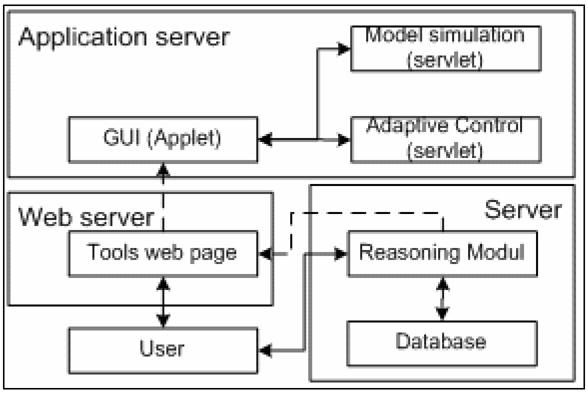


Figure 2
Example of the modules connection

## 2.1 Generic Block

The GB is the core of MARABU. This block consists of the autonomous agents group. Its task is to ensure the intelligent user interface, to transfer the information between the user and the distant server, where the external application starts. The GB's task is also to ensure the decision for the choice of the suitable method or tools according to the user's requirements. The Generic block consists of the following agents: UA, MA, CA, SA, and MoA.

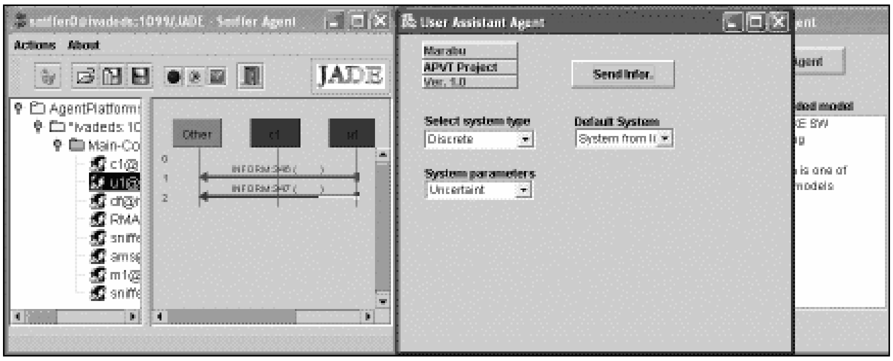


Figure 3
Prototype of the GB

The GB is developed in JAVA language (Figure 3); this ensures the GB's independence of the platform. The GB agents are developed using Java Agent Development (JADE) framework [17].

The JADE Platform simplifies the implementation of multi-agent systems by the form of middleware implemented in Java, which is compatible with the FIPA specification. The JADE Platform is not only an API for the development of multi-agent systems, but also includes a set of graphical tools (GUI) for the debugging, administration and user applications. Similarly, this platform supports the distributed configuration among computers with different operation systems. The configuration is made by a distant GUI. This GUI also offers a tool enabling agent migration among different computers. Thus, agents can migrate from one computer to another in any desirable time.

Three basic elements for the GB have to be defined:

**Attributes** – are defined within ontology as properties (slots) for each concept.

**Rules** – are defined within an inference engine and serves to find solution for the user on the basis of user's requirements.

**Control strategies** – are defined for searching solutions on the basis of the predefined rules. The known control strategies used for Case Based Reasoning (CBR) are: forward chaining and backward chaining.

CBR is used within the system to find a solution that matches best to the user's requirements. The CBR uses data stored in a database. The structured database is built independently for all three blocks of the systems[1]; however, all these databases are related to each other. The database contains a case library, and a set of solutions related to these cases. Objects in the database are represented by attributes. For an attribute-value representation a couple of methods can be used to measure the similarity degree. A method of weighed attributes is used to assess similarity between a current case (defined by the user) and a case in the case base. The weights allow to express the importance of all attributes.

Two different methods are used to evaluate attributes: user specific weights, which are given as a part of user's requirement through the user's questionnaire; and case specific weights (weights are specified as a part of structured database). The third option is to use combination of user specific and case specific weights to find the solution that fulfils the users' expectations best and does not neglect any important attributes given by case specific weights.

---

[1] There are three main blocks of systems, namely continuous dynamic systems, discrete event dynamic systems and distributed dynamic systems.

# 3 Methods and Tools

Methods and tools used in control theory to model, control and simulate various systems are arranged in the database. In order to build the database, the methods and tools had to be characterized by attributes, which determines their applicability from the aspect of users requirements.

The attributes are organized in three groups [2][9]:

1 Attributes, which are significant for different systems. This significance determines corresponding value for each attribute, according to these values the suitable method or a tool for the given task can be chosen.

2 Attributes, which are used by the corresponding methods. Such attributes characterize the concrete method which denotes that their utilization is not suitable for the other systems, i.e.it is not possible to give a suitable value for an other system.

3 Attributes, which characterize the prepared cases in detail and are stored in the database to be utilized in case based reasoning (CBR).

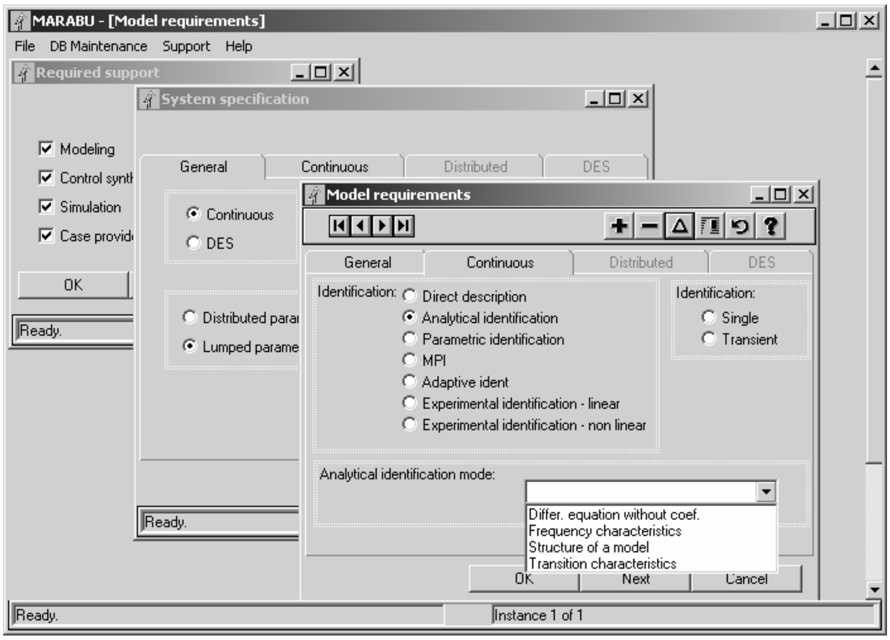


Figure 4
User interface of MARABU

The first prototype of the object-oriented database and the forms for data inserting and querying were realized in JADE environment [18]. For example, Figure 4 shows the graphical user interface of MARABU. In this figure, readers can see DB Maintenance interface that enables to insert and update the methods, tools and examples in database. Recently the database of the whole system is implemented on MySQL database server. The MySQL server has been chosen from a variety of database servers, because it is very flexible, fast, easy to administrate, and data can be accessed from various environments such as PHP, ASP.NET, Java, etc.

## 3.1 Ontology Utilization in MARABU

MARABU is created as a knowledge system. Knowledge representation is the application of logic and ontology to the task of constructing models for an application domain. Each of the three basic fields (logic, ontology, and computation) presents a different class of problems for knowledge sharing. In the MARABU system, knowledge is modeled using OWL – Web Ontology Language. The application of the OWL format for ontology in the agent system is relatively new. An advantage of OWL ontology is in the availability of tools that can reason about it. That tools provide generic support that is not specific to the

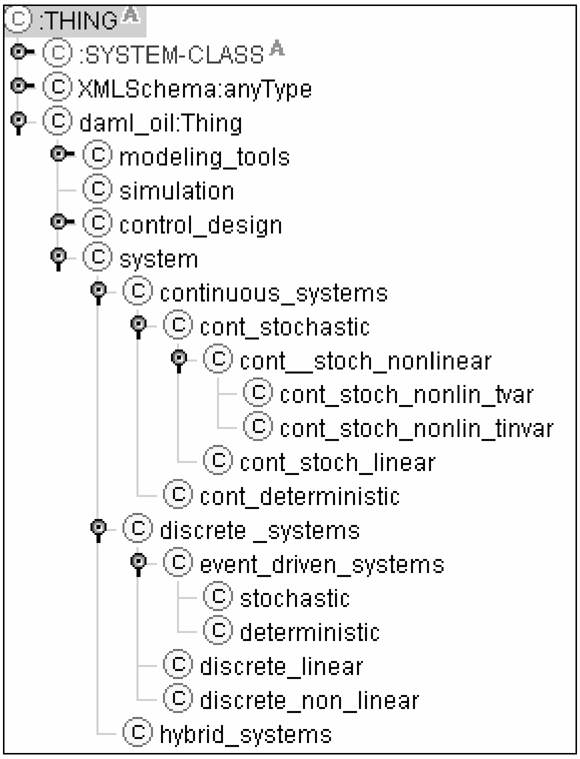


Figure 5
Ontology for MARABU using the PROTÉGÉ ontology editor

particular subject domain. Building a useful and reliable reasoning system is not a simple effort. Constructing ontology is easier. Constructing ontology in OWL enables to benefit from third party tools based on the formal properties of the OWL language.

OWL ontology is a sequence of axioms and facts, plus inclusion references to other ontologies, which are considered to be included in the ontology. The example of ontology for MARABU is shown in Figure 5.

## 4 Interface for External Application

A very important role of the GB (except monitoring of user\s requirements) is also to ensure the interface for the connection of external applications. shows a scheme of possible connection to the external application, which was realized in MARABU. On the first PC works the external application allowing, for example, the possibilities of modelling or control of a dynamic system; on the second PC the GB makes the above accessible on the web.

The external applications integrated in MARABU have to satisfy the following requirements:

- access via the internet
- starting as independent applications
- implementation of the interface
- the computing may be started by server

An example of possible external application is shown in Figure 6.

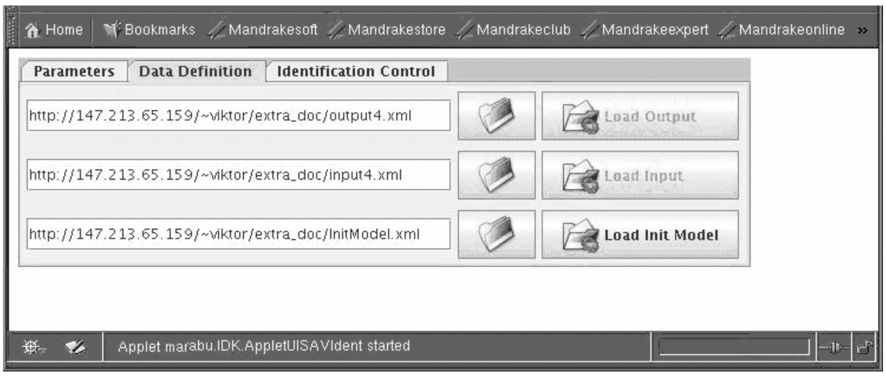


Figure 6
Example of the external applicaiton - definition of inputs and outputs for indentifiaction of controled system

## 4.1 Possible Future Development

The application of the MARABU system allows a grid architecture utilization. This architecture gives the possibility for the distributed computing via the internet or a local network. Although the resources are physically localized on different places, the access to all resources is available for the user. The grid architecture for MARABU is shown in Figure 7.

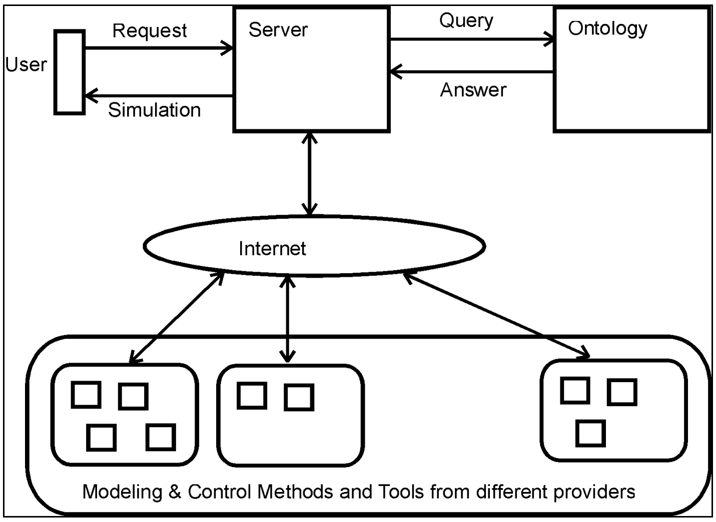


Figure 7
Grid plan of MARABU

**Conclusion**

In this paper the multi-agent support system MARABU is briefly described. The system is intended to serve in modeling, control system design and simulation of dynamic systems, either continuous or discrete, with strong emphasis on production systems. Therefore the system handles a lot of knowledge about methods, tools, and algorithms for modeling, control and simulation, and also about continuous and discrete event dynamic systems. The system also integrates work from many areas of control theory and of artificial intelligence. The system is large and distributed; it provides a good working environment for users (experts, students, technicians, etc.) to enrich their knowledge (either practical or theoretical) in the area of control theory.

**Acnowledgement**

This work was supported by Science and Technology Assistance Agency under contract No. APVT-51-011602.

**References**


[1] Bergmann, R.: Special issue on case-based reasoning, Engineering Application of AI, No. 12, 1999, pp. 661-759

[2] Budinska I., Dang T. T.: Ontology utilization in MARABU - a support system for modeling, simulation and control design. Proceeding of IEEE INES 2004, Edited: Nedevschi; Rudas, ISBN 973/662/120-0, 2004 p. 301-306

[3] Budinská I., Dang T. T.: A Case Based Reasoning in Multi-Agent Support System, Proceedings of the 6th International Scientific-Technical Conference PROCESS CONTROL, Kouty nad Desnou, Czech Republic, June 2004, 7 pages

[4] Budinská I., Frankovič B., Kasanický T., Dang T. T.: A Multi-agent Support System for Modelling, Control Design, and Simulation, Proceedings of IFAC Multitrack Conference on Advanced Control Strategies for Social and Economic Systems, September 2004, Vienna 6 pages (printed version will be available later this year as a Lecture Notes)

[5] Dang T. T.: Improving plan quality through agent coalitions, Proceedings of the 2nd IEEE International Conference on Computational Cybernetics, ICCC 2004, Eds: W. Elmenreich et al, August - September 2004, Vienna pp. 6, on CDROM ISBN 3-902463- 01-5

[6] Dang T. T., Frankovič B., Budinská I.: Case-based reasoning applied for CAS-decision system, Proceedings of the 2nd IFAC Conference: Control Systems Design, Š. Kozák, M. Huba (Eds.), Bratislava, September 2003, 6 pages, on CDROM

[7] Fogel, J.: Control Synthesis of Discrete Manufacturing Systems using Timed Finite Automata, Electrical and Computer Engineering Series, Advances in Signal Processing Robotics and Communications Edited by Kluev, Mastorakis, WSEAS Press, ISBN 960-8052-42-4 pp. 276-280

[8] Frankovič B., Budinská I., Dang T. T.: Cooperative Agents for Experience and Knowledge Management, Proceedings of the 3rd International Conference on Emerging Telecommunications Technologies and Applications ICETA 2004, September 2004, Kosice, Slovakia, pp. 205-211, Eds.: Frantisek Jakab et al, Editorial: Elfa, s.r.o., ISBN 80-89066-85-2

[9] Hrúz B.; Mrafko L.: Modelovanie a riadenie diskrétnych udalostných systémov s využitím Petriho sietí a iných nástrojov, Bratislava 2003, 1. vyd., Vydavateľstvo STU, 297 p., ISBN 80-227-1883-1

[10] Kozák Štefan: Toolbox PID- Virtuálna realita, Proceedings of 6th International Scientific - Technical Conference Process Control 2004, Kouty nad Děsnou , June 2004, R276

[11] Madarász L.: Inteligentné technológie a ich aplikácie v zložitých systémoch, University Press, Elfa 2004

[12] Liguš J. – Ligušová I, – Zolotová, I.: Remote and Virtual labs in Control Engineering Studies, Proceedings of 15th EAEEIE Annual Conference on Innovation In Education for Electrical and Information Engineering (EIE), Sofia, Bulgaria, 27-29 May 2004, pp. 193-196, ISBN 945-9518-22-1

[13] Sebestyénová J., Usage of Domain Ontology in e-Learning, Proceedings of 5th International Conference Virtual University VU'04, December 2004, Bratislava, 6 pages

[14] Végh, P.; Hulkó, G.; Belavý, C. Robust control of distributed parameter systems, Proceedings of 14th International Conference on Process Control 2003, June 2003, High Tatras, Slovakia

[15] http://www.w3.org/TR/2002/WD-owl-absyn-20020729/#3

[16] Fridman Noy N., Hafner C. D. The state of the art in ontology design, http://www.aaai.org/Library/Magazine/Vol18/18-03/Papers/AIMag18-03-006.pdf

[17] http://jade.tilab.com

[18] http://www.jadeworld.com