Design of the Agent-based Intelligent Control System

Baltazár Frankovič, Viktor Oravec

Institute of Informatics, Slovak Academy of Sciences
Dúbravská 9, 84507 Bratislava, Slovakia
utrrfran@savba.sk, upsyviki@savba.sk

Abstract: This paper introduces the possibilities of multi-agent system application for the modeling and intelligent control in the case of coarse ceramics burning process. It consists of technological description of this process, its decomposition into agents and macro-model of the decision system. Then multi-agent system modeling tools, such as alternating-time temporal logic and alternating transition systems and their epistemic extensions, are presented.

Keywords: multi-agent system, decision system, macro model, coarse ceramic burning production process, design

1 Introduction

For the creation of the intelligent control system (IDC) of production processes, which includes the decision process, suitable tools must be developed. From this point of view the research in this area is oriented to the development of algorithms which use the principle of artificial intelligence. The algorithms should satisfy the following three conditions:

- there is a natural object (in technical practices it could be, for example, different types of technological processes) with the properties of using the perception, decision, reasoning, etc.
- there is an aim to create its duplicate as, for example, the model of the natural object,
- there is a possible way for the realization of the supposed aim (the implementation of the concrete process by man or by some algorithm realized on computer).

The solution of the above problems (evolutionary algorithm [9], neural network [9], decision process [5] [7] [10], game theory [8], multi-agent system [5] [6] [7] etc.), is the subject of many publications. From our aspect it seems that for the creation of ICS for production system the MAS may have several advantages. MAS can be used to solve a complex problem as, for example:

- communication architecture provides the negotiation mechanism
- information architecture provides the framework for information modeling on negotiation.

The "mobile agent" based negotiation collects information and makes a decision for itself. It could be said that from logical/functional point of view, an agent-based distributed control system (IDCS) is a systematic network (within or without a hierarchy) of various local decision makers, which have independent knowledge sources such as database systems.

This advantage was, for us, the motivation to using the MAS, as the intelligent tool, for the solution of the decision process as a part of (IDCS) for COARSE CERAMIC BURNING PROCESS (CCBP).

The main scope of this paper is as follows: in section two the significant characteristics of MAS are described from the aspect of CCBP; in section three the CCBP technologies are formulated as an illustrative example of a multi-agent system. In section four a simple decision system was chosen as an application of multi-agent system. Then, the decomposition into agents is made. This section also contains macro-model of decision system. In section five, multi-agent design is discussed. In that section alternating-time temporal logic (ATL), alternating transition systems (ATS) and their extension are given. Section six contains the numerical results obtained from the verification of IDCS on the CCBP mathematical model.

2 The Significant Characteristics of MAS

Multi-agent system (MAS) may be considered as intelligent tool for the solution of such problems as planning, scheduling, decision making and control in the framework of production processes. The MAS approach seems to be the most feasible. It respects the complicated characteristics of the goal that we aim to achieve. There are some significant reasons that motivate us to choose the MAS approach to the solution of decision making, such as:

Modularity: Each agent is an autonomous module and can work without interventions of the external world. Each agent can have different capabilities or functionalities and through cooperation the agents are able to achieve a variety of goals.

Parallelism: The MAS approach enables to work in parallel. A complicated problem could be solved in an acceptable time by using a number of agents, e.g., gathering information from various resources allocated in different places.

Flexibility: The MAS approach is able to react in a flexible manner to each change in the environment. Through cooperation the agents can assist each other to compensate the lack of capability or knowledge. They can share information or own capacity to resolve a newly appeared situation, if one agent is not able to do so. Beside that, each intelligent agent can do reasoning about whom and when it has to cooperate with, in order to achieve the effective performance.

Of course, there are also some difficult questions associated with the MAS approach, e.g., which types of agents are needed, how many agents are optimal, what is a functionality of each agent, cooperation between agents, etc. We will have to successively deal with all these problems during developing this system, but in this paper we focus only on solving the problem how the agent system can assist the user in designing IDCS. We suppose that agents used in IDCS satisfied to the following properties:

- *Autonomy*: Each agent, as mentioned before, thinks and acts locally. It means that agent operates without direct interventions from other agents to achieve its own goals.
- *Social ability*: Agents can cooperate with other agents to achieve common goals.
- *Reactivity*: Agents react on changes in environment; it is needed to describe the negotiation process.
- *Pro-activeness*: Agents do more than response on events generated by environment, they can show goal-directed behavior.

3 Formulation of the Coarse Ceramics Burning Process Technologies – Illustrative Example

Our application is used to control temperature and gas flow in a tunnel kiln during coarse ceramics burning process [3]. The whole process is divided into three phases: heating, cooling and drying. Each product has to proceed through all phases of the process. The heating phase is the main one, where products are burned. After burning, the product proceeds into the cooling phase. When its temperature is the same as at the beginning of the heating phase, the drying phase can begin. This is the final stage, where residual humidity is evaporated.

Since big changes of coarse ceramics temperature could cause product damage, the kiln temperature is controlled in several zones. Each phase can be divided into zones that differ from each other, i.e. the principle of each zone is the same, but the parameters are different. The number of zones in the heating and cooling phase can vary from time to time, but the drying phase has always two zones. Due to the absence of product temperature measuring, the system has first to control temperature of the kiln and then to wait till the product is heated up to certain temperature. For this approach each zone has to be divided into two stages. The first stage is the temperature changing stage, i.e. the system changes the kiln temperature from one level to another. During the second stage the product is heated to the kiln temperature particular for the given zone, i.e. the system remains in this stage for a certain time. The set of zones is defined by the kiln temperature limiting curve (Figure 1). An example of such curve is shown in the next picture.

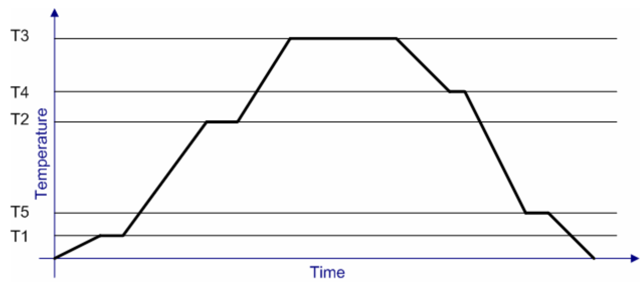

Figure 1
An example of limiting curve

The drying phase is always divided into two zones. During the first stage of the first zone the product is warmed up to the required temperature. Then residual humidity will be evaporated during the next stage and during the first stage of the second zone. During the third stage the product cools down.

Because of the kiln temperature limit curve, an adaptive algorithm with reference signal has to be used to control the temperature of the kiln. This adaptive algorithm is extended by the reference model algorithm, because of kiln parameters change. Due to changes of the kiln parameters, the kiln is "continuously" identified by some identification method.

Control of previously mentioned process is presently implemented with adaptive control algorithm an on-line identification [3]. This conventional approach works, but it is not so flexible as modern approach which is described in this paper. As mentioned in sections 1, 2 above, multi-agent system has several properties, which make the solution more flexible and intelligent. The conventional approach can be used for the described process only.

4 Design of a Decision System

Complete formalization of a decision model consists of two definitions:

- Definition of objective function – Objective function that has same meaning as fitness function in theory of genetic algorithms. Objective function helps decide which alternative in the decision process is the best. In our case, the objective function is the limiting curve.

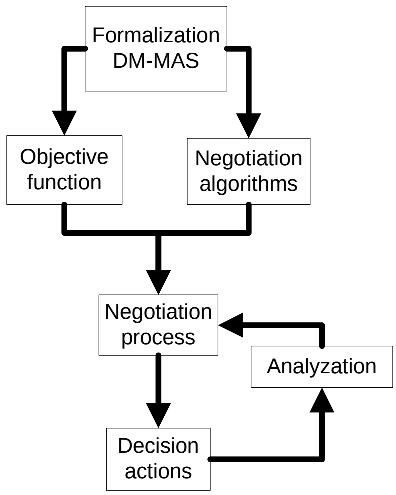

Figure 2
Design of decision system

- Definition of negotiation algorithm – The second definition in decision model formalization is the negotiation algorithm, which includes basic rules how to decide. This definition is included in the following three subsections. The first subsection, vertical cut, is the basic decomposition of the production process described in section 3. Vertical cut is something like domain analysis of the production process, i.e. basic entities are defined. The next subsection is the horizontal cut. More precise analysis is taken in the horizontal cut. Communication between entities is projected, negotiation algorithm is described. Finally, the last subsection present formal description of negotiation process.

4.1 Vertical Cut

The whole system is composed of five levels. The lowest level is the process level consisting of a *process model*, a *reference model* and a limit curve generation model (a *reference signal generator*). The next one is the level of control and identification algorithms. Note that previous levels are not included in the decision system.

The third (lower decision) level is composed of two decision models (DM), *Stages* and *Identification*. The *Identification* DM does not belong to hierarchical decision system (HDS), i.e. it stands and acts alone in the whole decision system, and its decisions do not depend on decisions made by other DMs. This DM decides whether a process of identification will start or not. The *Stages* DM is the lowest part of our hierarchical decision system. It decides whether the process will advance to the first or second stage.

The second (middle decision) level consists of one decision model - *Zones*. This model decides to which zone the process proceeds. This DM is the highest model in the HDS that makes the decision useful for the control algorithm.

Finally we approached the third decision level. The highest level is composed of one decision model - *Phases*. This DM can be called the observation global supervisor, because it has no influence on the control algorithm and is used only for observation and recognition of the process phases.

4.2 Horizontal Cut

In the previous section the vertical cut of decision system was described. This horizontal cut shows the states inside the DM, and information flows between these nodes in an oriented graph $G = (D, I \cup H, T)$ [1], where D is a set of all *decision nodes* and it is the conjunction of decision node sets of each data model in HDS. I and H denote *inter-level information flow* and *information flow* inside the *same level*, respectively. T represents the event

$$T : (X) \rightarrow (D \times D); X \subseteq I \cup H \quad (1)$$

and $T = T_C \cup T_{UC}$. (2)

The whole system behaves autonomously, so there is only one *controlled* event, namely is $T_C = f(H_{Start\,Proces}, u)$, where $u \in U$ is a control action generated by the object of an external interaction - *Start*. All other events are *uncontrollable* $T_{UC} = f(I, H)$.

[1] Note that for the above reasons Identification DM is excluded from next considerations

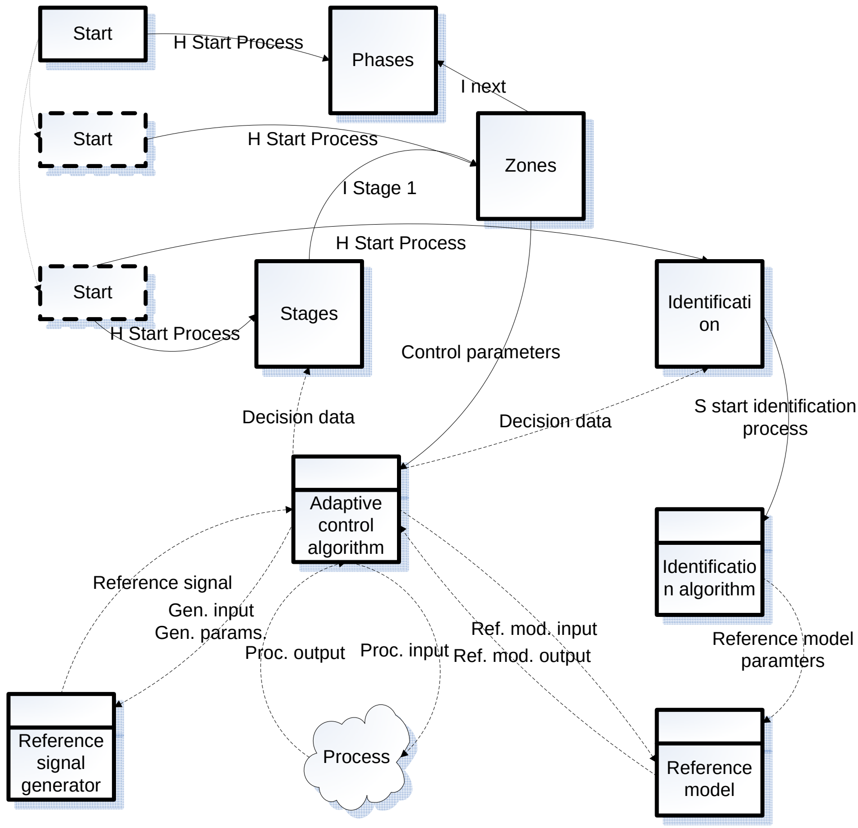

Figure 3

Vertical cut of decision system design

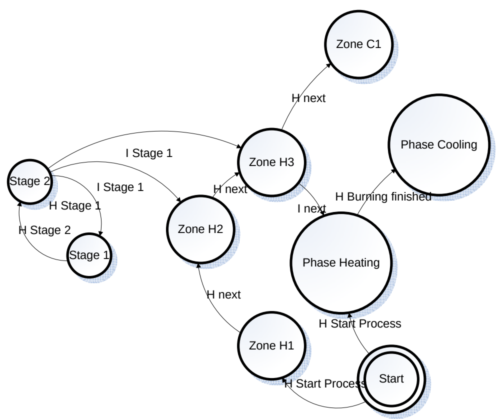

Figure 4

Part of design vertical cut

4.3 Formal Description of the Decision System

The previous considerations resulted in the following sets D, I, H, T, U :

$$D_P = \{D_{heating}, D_{cooling}, D_{drying}\}, \tag{3}$$

$$D_Z = \{D_{ZoneH1}, D_{ZoneH2}, ..., D_{ZoneHN_H}, D_{ZoneC1}, D_{ZoneC2}, ..., D_{ZoneCN_C}, D_{ZoneDC}, D_{ZoneDH}\}, \tag{4}$$

$$D_S = \{D_{Stage1}, D_{Stage2}\}, \tag{5}$$

$$I = \{I_{Stage1}, I_{Next}\}, \tag{6}$$

$$H = \{H_{StartProces}, H_{Stage2}, H_{Stage1}, H_{Next}, H_{HeatFinish}, H_{CoolFinish}, H_{DryFinish}\}, \tag{7}$$

$$S = \{S_{StartIdent}\}, \tag{8}$$

$$U = \{u\}. \tag{9}$$

where elements of D_P , D_Z and D_S are decision nodes of Phases DM, Zones DM and Stages DM, respectively. N_H , N_C stand for the total of decision nodes of the heating and cooling zones, respectively. S is a set of loop-back information and $S_{StartIdent}$ is the information that identification process has to be started. Set T is a set of events generated in the decision system, and each of them represents a mapping. Each of these mappings stands for relation between two decision nodes where the first is event generator and the other is event receiver.

$$T_1 \lfloor D_{heating}, D_{cooling} \rfloor \Rightarrow \exists H_{HeatFinish} \wedge \exists I_{Next} \tag{10}$$

$$T_2 \lfloor D_{cooling}, D_{drying} \rfloor \Rightarrow \exists H_{CoolFinish} \wedge \exists I_{Next} \tag{11}$$

$$T_3 \lfloor \lfloor D_{ZoneHj}, D_{ZoneHj+1} \rfloor \vee [D_{ZoneCm}, D_{ZoneCm+1}] \rfloor \Rightarrow \exists H_{Next} \wedge \neg \exists I_{Stage1} \\ j = 1,.., N_H - 1, k = 1,.., N_C - 1 \tag{12}$$

$$T_4 \lfloor \lfloor D_{ZoneHN_H}, D_{ZoneC1} \rfloor \vee \lfloor D_{ZoneHN_C}, D_{ZoneDH} \rfloor \rfloor \Rightarrow \exists H_{Next} \wedge \exists I_{Stage1} \tag{13}$$

$$T_5 \lfloor D_{Stage1}, D_{Stage2} \rfloor \Rightarrow \exists H_{Stage2} \tag{14}$$

$$T_6 \lfloor D_{Stage2}, D_{Stage1} \rfloor \Rightarrow \exists H_{Stage1} \tag{15}$$

5 Design of CCBP Multi-Agent System

In the previous sections, the problem of decision system design was transformed into that of multi-agent system design. Thus, let us take a look at multi-agent system design. Multi-agent system can be modeled in various ways, but here an approach is described that was invented for this technology. We present alternating-time temporal logic (ATL) and alternating epistemic transition systems (ATS). In general, two models are used in MAS modeling. The first model is MAS' "behavioral" model, where its behavior is described by ATL formulas. ATL formula is a mathematical formula that represents single systems' behavior, such as "Whenever heating h is finished, then the system will proceed into cooling phase in the next step." This formula can be written in alternating-time temporal logic like this;

$$\langle\langle\ \rangle\rangle \Box(\text{HeatingIsFinished} \rightarrow \langle\langle\ \rangle\rangle \circ \text{CoolingIsStarted}). \quad (16)$$

Note that in previous formula two propositions were used. Proposition HeatingIs Finished denotes that the system finishes the heating phase in the current step. Proposition CoolingIsStarted is true in the steps in which the system is in the cooling phase. The second model is MAS' "structural" model, where the structure of multi-agent system is captured in particular ATS. Parts of MAS structure are agents, states of agents, transitions between agents' states and propositions. Note that the set of propositions and the set of agents are the same in both models.

ATL and ATS are approaches which assume weak definition of agent, and this fits quite well for this application. But, if AI have to be modeled, epistemic extension of these approaches will have to be used. Epistemic extensions are alternating-time epistemic temporal logic and alternating epistemic transition systems. Agents' knowledge can be modeled in multi agent systems.

Using ATL and ATS or their epistemic extensions has several advantages. The most significant advantage is that the designer can use model checking algorithm to check the behavior of the designed transition system, i.e. to verify whether it is designed as intended. The designer's intention is described by set of ATL formulas.

6 Numerical Results from the Verification of Decision Process in CCBP Case

This section is intended to present practical results of our previous consideration. Experiments made in this section prove that our approaches are correct. Two decision systems can be found out in whole CCBP. The first one is the decision system of discrete event system (agents stages, zones and phases), let us label it as *discrete DS* (see section 4.1, 4.2) [4]. The second one is the decision system of adaptive control algorithm, *continuous DS* (6). Decisions made by discrete DS generate the following parameters:

- steepness of temperature change
- change of temperature (set point)
- time interval for stage 2
- error limit.

These parameters are different for each zone and are valid in whole zone, and are generated by agent Stages. Parameters are represented by loop-back information flow in the negotiation process. Inter-level information flow was represented by set of flows (4), but now it is better to use following function:

$$I : P \to M \,, \tag{17}$$

where M is set of messages and P is set of parameters, i.e. the function is message generator for agents communication.[2] Note that the parameters are symbols, which have value, thus they are not constants. Whole continuous DS consists of eight agents; one of them is the agent Stages is also included in discrete DS (see section 4.1). Agents are divided into to two levels.[3] Higher level consists of Adaptive control algorithm agent (Agent 7) and the agent Stages. As shown in Figure 5, the Stages agent generates messages I(R), I(W), which consist of particular stage parameters. Agent 7 makes the decision. Decisions are dependent on loop-back information flow from lower level ($S(Y_{RM})$, $S(Y_P)$, $S(Y_{NM})$). These decisions result in inter-level information flow, messages $I(U)$, respectively. This information is received by the agents Process (Agent 1), Model (Agent 2) and Reference Model (Agent 3). The first agent is the one which represents a process. It is precise identification of real process, i.e. in real application it is replaced by real process. The second one, Agent 2, is the agent which does not represent such precise identification of process as Agent 1. The last one, Agent 3, is the agent that represents a reference signal for adaptive algorithm. These three agents generate look-back information for Agent 7, as mentioned before. Look-back information flow can be represented as inter-level information flow by the function

$$S : P \to M \,. \tag{18}$$

Items of set P are generated by the following equations:

$$Y_{NM}(n+1) = 1.52359Y_{NM}(n) - 0.6362Y_{NM}(n-1) + 0.02476U(n) \tag{19}$$

$$Y_{RM}(n+1) = 0.95Y_{RM}(n) + 0.05U(n) \tag{20}$$

$$Y_P(n+1) = 1.5236Y_P(n) - 0.6364Y_P(n-1) + 0.0248U(n) \tag{21}$$

---

2 Inter-level information flow can be a unfinite set of elements, so it is better to describe it as function.

3 Agents are divided into levels with regard to: “Think globally, act locally”.

$$U(n+1) = 40.387722(Y_{NM}(n) + 0.05W(n) - Y_P(n) + ...$$
$$+ 0.95Y_{RM}(n) - 1.52359YNM(n) + 0.6362Y_{NM}(n-1)) \quad (22)$$

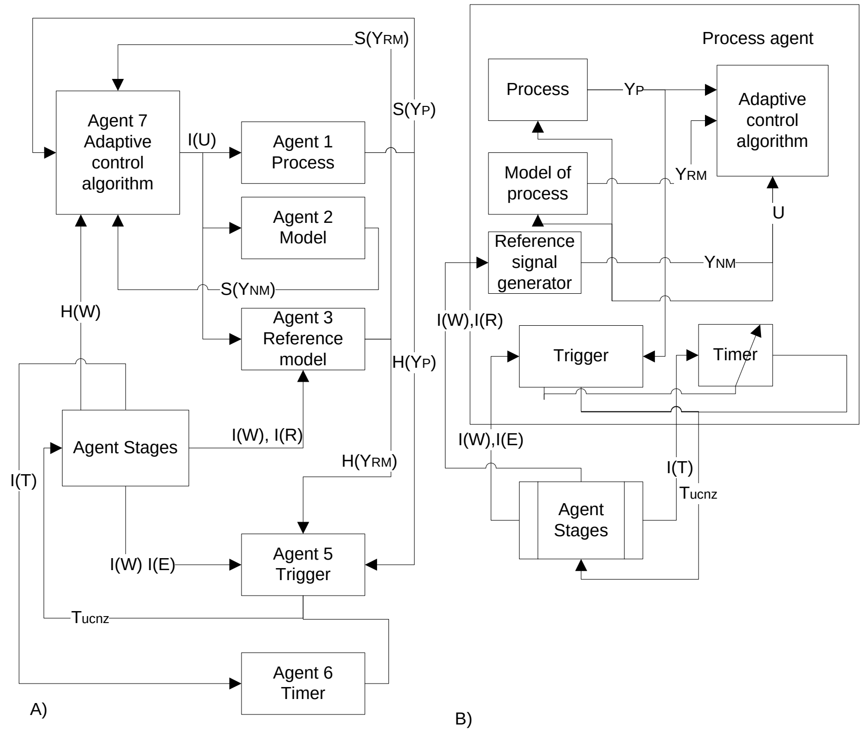

Figure 5

Description of agent based intelligent control system (A) and classical adaptive control system (B)

Two agents – Trigger (Agent 5) and Timer (Agent 6) are involved in continuous DS. These agents generate decision data through look-back information flow, as described in the scheme of vertical scheme (see section 4.1). This event represents the information that the Stage agent can proceed into next stage (1). The same event can be generated by two different agents, thus it has two definitions – one for agent 5, which decided upon level information flows, and one for agent 6, which decided upon time. Mathematical definition of the event is as follows:

$$T_{UCNZ}\left(H\left(Y_{RM}\right), H\left(Y_P\right)\right) = (D_5, D_{Stage1}) \quad (23)$$

and $T_{UCNZ}(I(T)) = (D_6, D_{Stage2})$, (24)

where D_5 and D_6 represent the decisions nodes of agents 5 and 6. D_{Stage1}, D_{Stage2} have been defined in previous sections (4.3)(5).

Our experiment includes simulation of zone parameters setup is as follows:

Table 1

Values of zone's parameters

Parameter of particular zone	*Value of parameter*
steepness of temperature change	0.5 steps
change of temperature (set point)	40°C
time interval for stage 2	80 steps
error limit	0,23°C

Parameters setup results in simulation, which includes 184 simulation steps. First 104 simulation steps are concluded into stage and this is the transition part of the control process. Error limit is satisfied in step 104, thus the system proceeds into next stage and remains in it for next 80 steps. Values of signals are shown in graph.

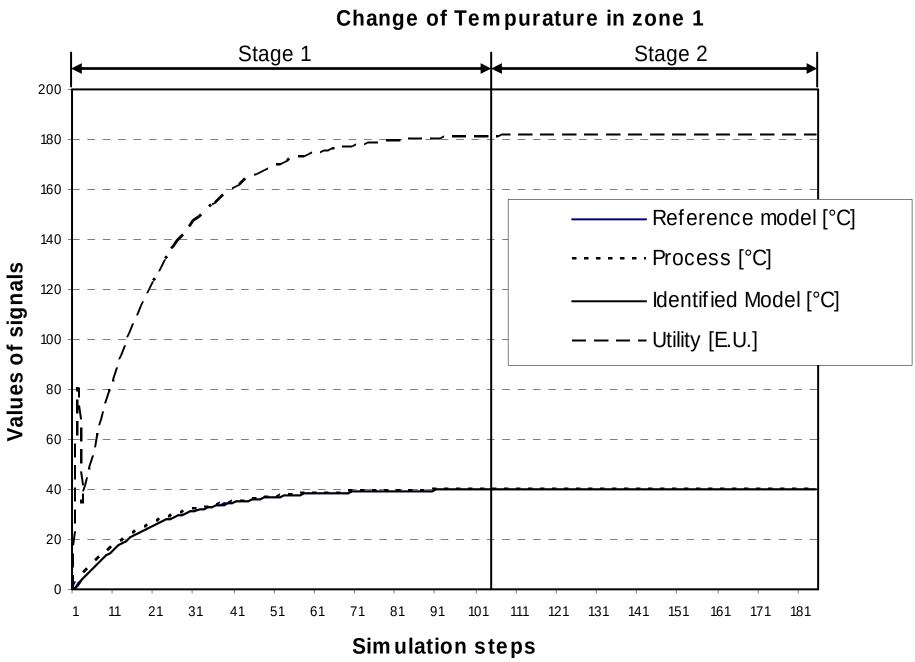

Figure 6

Graphical representations of experiments results

One can analyze from Figure 6 that control process is stable throughout the whole zone. Quality of control is evaluated by sums of error's square (SES) between signals. SES's between signals are shown in the following tables. One can see that adaptive algorithm is quite good because its uses the model of process parameters to evaluate utility. The process output follows the reference signal, but with bigger error than the model output. This is caused by the absence of continuous identification in the experiment.

Table 2
Relations between system signals Process, Model and Reference model

Signals	*SES*
Reference model, Model of Process	0.0023159
Reference model, Process	40.72941
Model of Process, Process	41.27523

Knowledge from the decision process verification

The experiment shows that multi-agent system can be used to control continuous dynamic systems. This approach, switching parameters of control algorithm, is similar with gain scheduling well-known in non-linear theory, but multi-agent system are used here. Multi agent systems are more flexible in structure and behaviour, and can be easily advanced with knew features. For instance, agent process can make some reasoning about adaptive and identification algorithm, type of process model, and more. These features can be implemented as separate instances and process agent only switches between them. In another case, multi agent system structure is so flexible that it can be customized to another similar application.

Acknowledgement

The work presented in the paper was supported by following projects:

- APVT – 51 – 011602
- VEGA 2/4148/24

Conclusion

The main scope of this paper was to introduce an illustrative example of an application for designing a multi-agent system. This paper presents an application of decision system. The decision system was decomposed into agents, modeled by macro model of control. However, this paper did not aim at designing a multi-agent system, although tools for designing such system were presented.

Macro model of control for a decision system is a quite simple and efficient approach to model decision system. It describes decision and data flow in the whole system. Decisions are modeled by events and data by response on events or by information flow. Then this model can be implemented in multi-agent technology, where agents communicate with each other by messages. Messages in multi-agent system represent events in macro model. Data flow in macro model can be represented by knowledge in multi-agent system. This is very crucial moment, because decision system can be modeled by multi-agent system; a multi-agent system can be modeled and verified by help of alternating-time temporal epistemic logic and alternating epistemic transition logic.

References

[1] M. Wooldridge, N. R. Jennings, “Intelligent Agents: Theory and Practice”, Knowledge Engineering Review 10(2), 1995

[2] M. Wooldridge, N. R. Jennings, “Pitfalls of Agent-Oriented Development”, Agent ’98: Proceedings of the Second International Conference on Autonomous Agents ACM Press, May 1998

[3] B. Frankovič, „Adaptívne a učiace sa systémy riadenia“, Veda, 1982

[4] V Oravec., J. Fogel, „Cookbook of alternating time systems“, 7th Scientific Conference on Electrical Engineering and Information Technologies for PhD. Students, 3 pages, February 2005

[5] I. Budinská, T. T. Dang: A Case Based Reasoning in a Multi Agents Support System, In Proc. of the 6th International Scientific –Technical Conference, Process Control 2004, June 2004

[6] I. Budinská, T. T. Dang: Ontology Utilization in MARABu – a Support System for Modelling, Simulation and Control Design, In Proc. of the International Conference on Intelligent Engineering Systems, INES 2004, September 2004

[7] T. T. Dang: "Improving plan quality through agent coalitions", IEEE International Conference on Computational Cybernetics – ICCC’04, 6. pages, 2004

[8] Sjoerd Druiven: “Knowledge Development in Games of Imperfect Information”, Master Thesis, Institute for Knowledge and Agent Technology, University Maastricht, May 2002

[9] Mařík V., Štěpánková O., Lažanský J., „Umělá inteligence (3)“, Academia, 2001

[10] Budinská I.: Generovanie trajektórie dynamickoho systému diskrétnych udalostí pomocou GSMP – Perturbačná analýza a jej použitie na optimalizáciu DEDS, Journal AT&P 8/97, 4 pages, august 1997