Journal of Cybernetics and Informatics

published by

Slovak Society for
Cybernetics and Informatics

Volume 5, 2005

http://www.sski.sk/casopis/index.php (home page)

ISSN: 1336-4774

Journal of Cybernetics and Informatics 5 (2005) 32-36
http://www.sski.sk/casopis/index.php

MULTI AGENT SYSTEMS SUPPORT DECISION SYSTEMS

Oravec, V., Frankovič, B.

Institute of Informatics, SAS, Dúbravská 9, 84507 Bratislava, Slovakia
upsyviki@savba.sk, utrrfran@savba.sk

Abstract: *This paper introduces an illustrative example of multi-agent system application. It consists of description of application, its decomposition into agents and macro model of 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, burning production process, design*

1 INTRODUCTION

Since researchers started thinking of artificial intelligence, they have been finding out how single cognitive entity can be modeled. Multi-agent system solves this problem quite well. Each agent has incomplete knowledge, computing resources and perspective. This is known as bounded rationality. Agents can create some systems, multi-agent systems. However, one agent thinks and act locally and whole system acts globally. Whole multi-agent system tries to achieve common goals. The word "agent" is often used in this paper, but what does it mean exactly? There is not exact definition of agent, but we can give you some ideas that are helpful in imagining an agent. In their works, Wooldridge and Jennings distinguish two definitions of agents: weak and strong definitions [1] [2]. Weak definition defines an agent as some hardware or (more usually) software-base computer system that satisfies 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;
- pro-activeness: agents do more than response on events generated by environment, they can show goal-directed behavior.

For example, weak definition leads to a server service that satisfies the previous properties. Stronger definition of an agent describes the agent as software computer system, which satisfies the above properties, but it is enriched with mental characteristics, such as knowledge, belief, intention and obligation. Stronger description is usually used by researchers working with artificial intelligence. Note that weak description is used throughout paper.

As mentioned before, multi-agent system is open system built up from agents. Complex problems can be solved by multi-agent system, where agents cooperate with each other to achieve the solution. It is usually a heterogeneous system, i.e. it consists of agents that are not of the same kind.

The main scope of this paper is to describe an illustrative example of a multi-agent system. Simple decision system was chosen as an application of multi-agent system. In the first section, this one, an introduction into multi-agent system technology is made. Then we proceed in section two, where an illustrative example is introduced. After that, the decomposition into agents is made in section three. Section three also contains macro model of decision system. In the fourth section, multi-agent design is discussed. In that section alternating-time temporal logic (ATL), alternating transition systems (ATS) and their extension are introduced.

2 COARSE CERAMICS BURNING PROCESS – 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 stages (zones). Each phase can be divided into zones that differ from each other, i.e. the principle of zones 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 zone 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. An example of such curve is shown in the next picture.

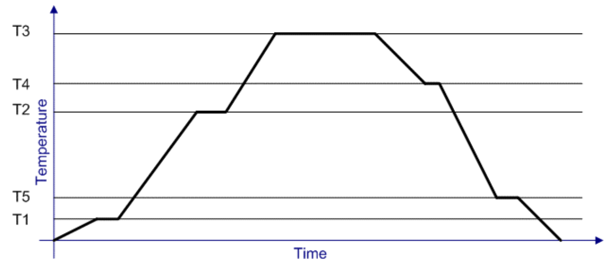

Figure 1. An example of limiting curve

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.

3 DESIGN OF A DECISION SYSTEM

Design of a decision system is projected by vertical and horizontal cuts. The vertical cut is used to present a decision system such as MAS, and, on the other hand, the horizontal cut presents the system as a state space machine.

3.1 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. *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.

3.2 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 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

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

event $T:(X)\rightarrow(D\times D); X\subseteq I\cup H$ and $T=T_C\cup T_{UC}$. 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* $T_{UC}=f(I,H)$.

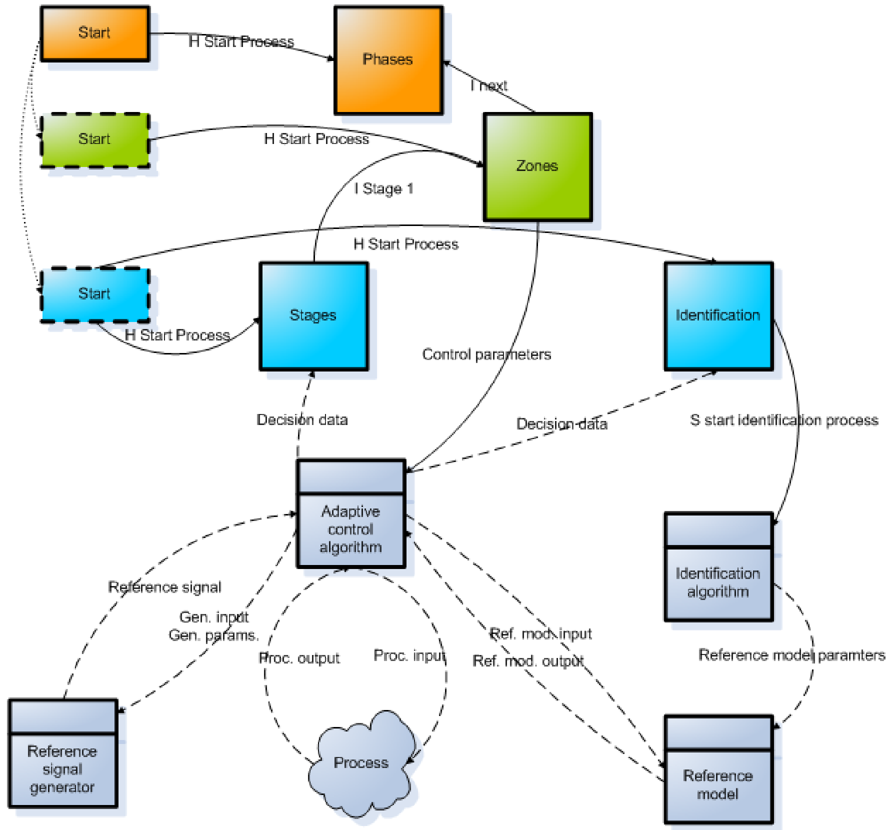

Figure 2 Vertical cut of decision system's design

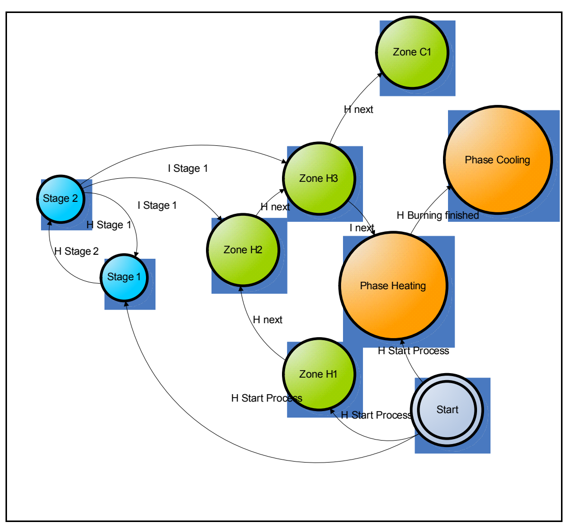

Figure 3 Part of design's vertical cut

3.3 Symbolic description of decision system

As result of previous consideration are following sets D, I, H, T, U :

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

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

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

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

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

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

$$U = \{u\}.\,.$$

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

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

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

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

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

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

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

4 DESIGN OF A MULTI-AGENT SYSTEM

Problem of decision system design was transformed into problem of multi-agent system design in previous sections. 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 phase heating 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,

$$\langle\langle\ \rangle\rangle \square(\text{HeatingIsFinished} \rightarrow \langle\langle\ \rangle\rangle \circ\text{CoolingIsStarted}).$$

Note that in previous formula two propositions were used. Proposition HeatingIsFinished denotes that system finishes the heating phase in the current step. Proposition CoolingIsStarted is true in that steps, in which the system is in the cooling phase. The second one 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 set of propositions and set of agents are the same in both models.

ATL and ATS are approaches which assume weak definition of agent and for this application it fits quite well. 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 was intended. The designer's intention is described by set of ATL formulas.

5 ACKNOWLEDGEMENT

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

- APVT – 51 – 011602
- VEGA 2/4148/24

6 CONCLUSION

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

Macro model of control for 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.

7 REFERENCES

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

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

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