Process and Logic Approaches in the Intelligent Agents Behavior

Michal RADECKÝ, Petr GAJDOŠ
Department of Computer Science, VŠB-Technical University of Ostrava
tř. 17. listopadu 15, 708 33 Ostrava-Poruba
Michal.Radecky@vsb.cz, Petr.Gajdos@vsb.cz

Abstract. The *Multi-Agent System (MAS)* technology is one of the possibilities of development of modern, powerful and advanced information systems. In the case of the multi-agent systems, some of standard approaches could be used, however they have to be adjusted or extended. This paper describes the ideas and methods for MAS modeling and developing, based on the internal agent behaviors and process modeling. This paper is concerned with the problems of behavior specification and reconfiguration. Our method is based on the process and logic approaches.

1 Introduction

The Multi-Agent System (MAS) technology is formed on the concepts of the Complex Systems (e.g. macromolecules, ants' colony, and economical systems) and also on the facilities and capabilities of software information systems [2], where the essential of MAS properties are autonomy and intelligence of elements, communication among elements, mobility of elements, decentralization of the control, adaptability, robustness, etc.

The MAS can be developed as a general information system that is composed from a number of autonomous elements (called *Agents*) [8]. In this context, the Multi-Agent System is a framework for agents, their lives, their communication and mobility. The *Agent* is a software entity, within the framework, created in order to meet its design objectives that are subordinated autonomously with respect to the environment, sensorial perceptions and internal behavior and also to the cooperation with other Agents.

2 Modeling of the Agent Behavior

Each Agent is determined by its own objectives among others. The ways to meet these objectives are founded on the internal behavior of this Agent. The internal behavior of each Agent is specified by the processes which express the algorithms of behavior [4]. The Agent lives, behaves and reacts to stimulus and to the environment, in accordance to the requirements and states of the internal behavior.

It is necessary to take into account the fact that each Agent is an absolutely autonomous element of MAS and thus the internal behavior have to constructed only from the processes, activities, knowledge and facilities that belong to a given Agent. Then, the consequent behavior of whole MAS is formed by *communication* and *cooperation* of separated Agents and their behaviors. This interaction is realized by the usage of *message passing* that is adapted to the demands of MAS.

2.1 Behavior Modeling by UML and its Extension

The UML (Unified Modeling Language) is an essential tool for process modeling, both on the business level and analytic level of description [6, 7]. It can be applicable for modeling of the internal behavior of the Agents as well. The *UML Activity Diagrams* are a standard diagrammatic technique that describes the series of activities, processes and other control elements that together express the algorithm. They are especially suitable for modeling of agent behavior; however, some modifications and extensions are required.

The forenamed extensions are implemented by the *Agent Behavior Diagrams (ABD)* which could contain all of the standard UML Activity Diagram elements, and some new elements are defined likewise for the modeling purposes of the agent processes. These new elements are concerned with the message passing among the Agents or with the other specific attributes of MAS or its elements. In early phases of MAS development process, these extensions are provided by the implementation of special "**send/receive activity nodes**" which include additional information about messages content and messages receiver/sender identification, see figure 1. The decision nodes coming from the standard UML Activity Diagrams are improved too. The modified "**decision nodes**" and their output edges can hold some extra information that is usable for the next control flows determination based on the incoming messages. This control flows are selected according to the agent's objectives.

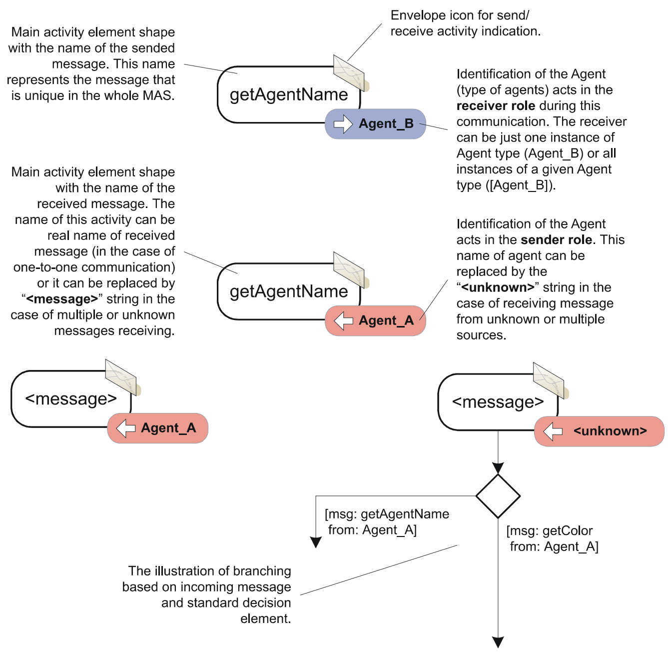

Figure 1: The examples of the new communication activity nodes and illustration of the extended decision node.

In connection with the example of new nodes depicted on figure 1, it is important to say follows. The term *Agent* expresses only the "type of agents" in the context of MAS modeling. The real separated Agents are the instances of this type, it is similar to relation between Class and Objects from Object Oriented Approaches. The particular Agents are not the issue of MAS modeling or design phases. They will appear not until in the implementation, simulation or operation phases.

2.2 Description of MAS Model

The basic terms and relations within the MAS Model are defined in this section. Whole model consists of six basic elements and it is quite similar to the Business Process Model (BPM) [3, 5, 6]. Some new components, like *realizations* or *messages*, are there as well.

Our "*MAS Model"* can be described by following n-tuple (A, O, P, R, Ac, M), where:

A	- is a finite set of *Agents*. Each Agent is defined by the name.
O	- is a finite set of *Objects*. They are also defined by the name. There are two basic types of objects depending on the usage within the atomic activity (Ac): 1. *Input Objects* - O_i 2. *Output Objects* - O_o , then $O = O_i \cup O_o$
P	- is a finite set of *Processes* or *Sub-Processes* of the whole MAS Model. Each process, except the Agent Primary Process, is specified by one or more realizations. Each process contains the name, owner (Agent) and the sets of input and output objects.
R	- is a finite set of *Realizations*. The realization presents a sequence of Activities, where the Sub-Processes are substituted by their diagrams (Agent Behavior Diagrams). Each Realization is defined by unique name and has two sets of input and output objects.
Ac	- is a finite set of *Activities*. Each Activity has unique name, one set of input and one set of output objects. Each activity can have assigned some value that expresses the time demands, costs, resources, etc.; this value is called Score. Two types of activities can be found in our model: 1. *Simple Activity* Ac_s – is the "standard" atomic activity, well known from Business or Software modeling. 2. *Communication Activity* Ac_c – is the message passing activity with the link to the one or more messages according to the particular usage (sending or receiving). , then $Ac = Ac_s \cup Ac_c$
M	- is a finite set of *Messages*. The Message is defined by unique string. It can contain a data segment and it is also able to hold some information about Agent who sends or receives it.

Then some relations can be found in the model as well:

r_1	- is a relation between Processes and Realizations. Each Process can be realized by several ways. $r_1 \subseteq (P \times R)$
r_2	- is a relation between Processes and their Activities. $r_2 \subseteq (P \times Ac)$
r_3	- is a relation between Activities. It specifies all sequences of the process control flows. $r_3 \subseteq (Ac \times Ac)$ Then the set R meets following condition: $R' \subseteq r_3$, then $R = R' \cup (\varepsilon , a_1) \cup (a_2 , \varepsilon)$, where $a_1, a_2 \in Ac$. Then a_1 is the initial activity, a_2 is the last activity of realization control flow and ε is an initial or final node (it is just diagrammatic node) of a given control flow.

r_4 - is a relation between Activities and Objects

$$r_{4i} \subseteq (O_i \times Ac)\ ,\ r_{4o} \subseteq (Ac \times O_o)\ ,\ r_4 = r_{4i} \cup r_{4o}$$

r_5 - is a relation between Realizations and Objects

$$r_{5i} \subseteq (O_i \times R) \mid \text{where for all } (o , r) \in r_{5i} \text{ there exists } (o , a) \in r_{4i} \text{ such that } (a , x) \text{ or } (x , a) \subseteq r \text{ and } a,x \in Ac,\ o \in O_i ,\ r \in R.$$

$$r_{5o} \subseteq (R \times O_o) \mid \text{where for all } (r , o) \in r_{5o} \text{ there exists } (a , o) \in r_{4o} \text{ such that } (a , x) \text{ or } (x , a) \subseteq r \text{ and } a,x \in Ac,\ o \in O_o ,\ r \in R.$$

$$\text{Then } r_5 = r_{5i} \cup r_{5o}$$

r_6 - is a relation between Processes and Objects.

$$r_{6i} \subseteq (O_i \times P) \mid \text{where for all } (o , p) \in r_{6i} \text{ there exists } (o , r) \in r_{5i} \text{ such that } (p , r) \subseteq r_1$$

$$r_{6o} \subseteq (P \times O_o) \mid \text{where for all } (p , o) \in r_{6o} \text{ there exists } (r , o) \in r_{5o} \text{ such that } (p , r) \subseteq r_1$$

$$\text{Then } r_6 = r_{6i} \cup r_{6o}$$

r_7 - is a relation between Communication Activity and Messages.

$$r_7 \subseteq (Ac_c \times M)$$

3 Intelligence within the Agent Behavior

Above mentioned modeling is described from static and structural point of view only. Though, it is necessary to speak also about dynamical aspects which brings the term *Intelligent Agent*. It is a standard Agent that disposes of certain kind of "brainpower". This capability is hidden inside the agent behavior and it can be found in various points of behavior algorithms. The intelligence can be ensured by the application of several tools, e.g. logic, artificial intelligence, expert system, etc.

The intelligence within the agent behavior can be concerned with three points:

- *Intelligence contained in the Activities* – it is a problem of activities implementation phase and it is hidden from the modeling perspective. Only the results of this activity firing are relevant. The task for the logic is to make decisions and derivate new information within the activity, e.g. weather forecast.
- *Intelligence of the control flow routing* – this application of logic or intelligent tools is covered in the decision points. The decision making and deduction of next behavior are activated whenever the "intelligent decision point" is reached during the process realization execution. The intelligent control flow routing can be used for all branching that request more complex and knowledge-based decision making, e.g. suitable car to a given cargo assignment. The control flow branching is based on the results of a given decision point and on its output edges and conditions.
- *Intelligent selection of Process Realization* – the third task of intelligence subsumed into the Agent life is concerned with the real-time operation of MAS. The potential and possibilities of this application is mentioned in the next chapter.

3.1 Intelligent Selection of Process Realization

Each Agent must try to realize its tasks and solve the upcoming situations in order to meet its design objectives. From this point of view, the standard Agent is grounded on the finite and constricted description of its behavior that is already defined during the modeling phase of the MAS. Therefore, there is no way to change the behavior during the Agent execution and life. In the case of *Intelligent Agent* it is able to do this. This kind of Agent can dynamically change some parts of its own behavior according to the situations and environment. This principle is denoted as *behavior reconfiguration approach*. The reconfiguration approach is founded on the replacement of a given part (Sub-Process) of the Process by another one that is the most suitable for current situation and conditions. The Realizations represent all possible ways to perform a given Process. The important and expected situation will appear whenever one Process is aggregated from two or more Realizations. The logic tools are responsible for solving this situation.

According to these ideas, it is able to define such procedure as an algorithm of reconfiguration process:

1. *The Specification Phase* - definition of all Realizations related to the Process that could be reconfigured.
2. *The Selection Phase* - checking of the applicable Realizations of a given Process and finding of the most suitable one.
3. *The Execution Phase* - chosen Realization firing.

3.1.1 Selection Algorithm and Illustrative Example

The figure 2 shows the basic scheme of our method. This selection is concerned with two steps included in the second step of the reconfiguration algorithm (the Selection Phase). At the beginning, we already have specified a set of all Processes and their Realizations (the Specification Phase). The number of Processes in the MAS Model depends on the author of this model and on the complexity of whole system. The main idea of our method consists of two steps within the Selection Phase.

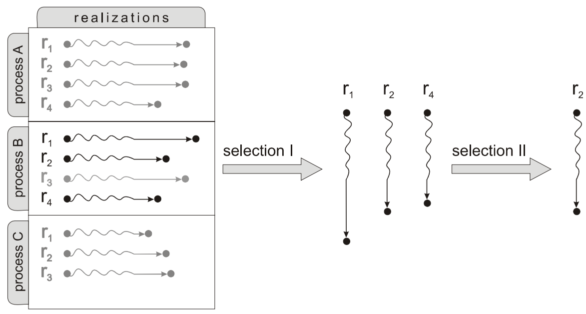

Figure 2: The basic scheme of our method

Selection I (selection of applicable realizations): This selection is based on the facilities of process approach. First, the Process that will be reconfigured is selected (see `Process B` in the figure 2). This Process has to finish with required state and with required output objects as well. The selected Process should be done by four Realizations. However, some of these realizations could not be executed. Therefore, the output of the **Selection I** is a subset of all applicable Realizations of selected Process. Each element of this set can have assigned some indicators like time, costs, etc. These indicators are called *Scores of Realization*. The score is based on the control flows and on the scores of activities within it.

Selection II (selection of the most suitable realization): The second step of our selection method is based on the logic approach. Till now, the set of Realization is known now. They could be executed inside the MAS as a given Process. The simply question occurs: Which is the best one? The decision is based on many parameters owned by activities, agents or objects. All of these parameters have to be specified within the MAS modeling phase. A predicate, fuzzy logic, Transparent Intensional Logic (TIL) or some other approaches e.g. the Formal Concept Analysis could be used for finding of the most suitable and effective Realization.

4 Future Work

Till now, the meta-model of MAS, its elements and relationships have been specified. However, only the theoretical conclusions are not sufficient themselves. A real application has to be created to demonstrate theoretical results in practice. Now, the application called "AgentStudio" is ready for use. It makes it possible to specify Agent behavior with all above mentioned extensions. Next step will consists of the MAS Model analysis based on known approaches, e.g. on the Formal Concept Analysis, Cluster analysis etc. We want to use JADE framework and generate agents' source codes or their templates directly from AgentStudio application. The main goal of our future work is to create software, that will be based on the clear methodology and that will allow to specify whole MAS without thorough knowledge of MAS framework.

5 Conclusion

This paper is concerned with the MAS technology especially with the specification of the Multi-Agent System Model. These specification tools must be designed with a respect to the skills of standard users that will be a "modelers" of the MAS and that will determined the objectives, requirements and behavior of whole MAS from the real-world point of view. Extended UML modeling approach was mentioned in this paper. Some new modeling elements and their graphical representation were defined. New application called "AgentStudio" was introduced for these purposes as well.

This paper has been supported by the program "Information Society" of the Czech Academy of Sciences, project No. 1ET101940420 "Logic and Artificial Intelligence for multi-agent systems"

6 References

[1] Aalst, W.M.P. van der. 1997. "Verification of Workflow Nets". In Application and Theory of Petri Nets 1997, P. Azema and G. Balbo (Eds.), Springer-Verlag, Berlin, 407-426

[2] Kubik, A. 2004. "Intelligent Agents", Computer Press, Prague, (publication in czech language).

[3] Radecky, M. and Vondrak, I. 2005. "Formalization of Business Process Modeling". ISIM 2005, Hradec nad Moravici.

[4] Radecky, M. and Vondrak, I. 2005. "Modeling of Agents Behavior within MAS". Information systems in practice 2005, MARQ, Ostrava, (paper in Czech).

[5] Radecky, M. and Vondrak, I. 2005. "Modeling of Processes". WOFEX 2005, Ostrava.

[6] Vondrak, I. 2004 "Methods of Business Modeling", VSB-TUO, Ostrava, (lecture notes in Czech).

[7] Bauer, B. and Muller, J.P. and Odell, J. 2001 "Agent UML", Springer-Verlag, Berlin

[8] Muller, J.P. 1996 "The Design of Intelligent Agents. A Layered Approach". Springer-Verlag, Berlin