Intelligent Selection of Realizations within the Agent Behavior

Michal Radecký
Faculty of Electrical Engineering and Computer Science
VSB-TU Ostrava
17.listopadu 15, Ostrava, Czech Republic
e-mail: michal.radecky@vsb.cz

KEYWORDS

Agent, Intelligence, UML Activity Diagram, Process, MAS Model, MAS Meta-model, **Agent**Studio

ABSTRACT

The importance and attraction of the *Multi-Agent System (MAS)* topic are increased, nowadays. The techniques for the modeling of such systems are indivisible part of their software processes. They have to be applied with emphasis on the special requirements or properties of Multi-Agent Systems, e.g. autonomy of elements, intelligence of elements. This paper describes the ideas and methods to model and develop Intelligent MAS, based on the internal agent behavior, its reconfiguration, and intelligent selection of the most suitable or applicable Process Realizations. The fundamental elements of our MAS Model, UML Activity Diagram extensions or intelligence within the Agent's life are mentioned in this paper.

INTRODUCTION

The MAS technology is based on the concepts of the Complex Systems (e.g. macromolecules, ants colony, economical systems), and on the facilities and capabilities of software information systems (Kubik 2004) which are taken into account during the analysis, design and implementation of a given MAS.

The MAS can be formed as an general information system that is composed from a number of autonomous elements (called *Agents*). In this context, the Multi-Agent System is a framework for Agents, their lives, their communication and mobility and it is an environment where the goals of the particular Agents should be obtained too.

The *Agents* are such elements of MAS. They are the software entities created in order to meet their design objectives. These objectives are subordinated autonomously with respect to the environment, sensorial perceptions, internal behavior and to the cooperation with the other Agents.

AGENT WITHIN THE MAS

The several types of Agents could be found in one Multi-Agent System as well as in the "real world" which is realized by this MAS. These "Types of Agents" are able to denote as *Agent Classes*, according to the goals, internal architecture and behavior of particular Agents. The combination of two perspectives is concerned with the agent's classification - outside behavioral view (proactive or reactive agents) and the view of internal specification (deliberative or process specifications) (Radecky and Vondrak 2005). The classes that describe a process approach in the combination with reactive or proactive perspectives are the main goal of this work and research. These classes cover the concept of "Intelligent Agents" that is formed on the principles of an internal agent behavior.

Each Agent is determined by its own objectives and the way to meet these objectives is founded on the internal behavior of a given Agent. Internal behavior of such Agent is specified by the algorithms. The Agent lives, behaves and reacts to stimulus and environment, according to the requirements of the algorithms. Any Agent in the MAS has the main internal life Process, called *Primary Process*, consequent on the Agent's classification. This Primary Process can be decomposed into a number of Sub-Processes that refine the internal behavior on.

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 be based only on the Processes, Activities, knowledge and facilities that belong to a given Agent. Then, the destination behavior of whole MAS is formed by communication of separated Agents and by interconnection of several internal agent behaviors. This interaction is realized through the use of message passing adapted to the demands of MAS.

In the context of MAS modeling and agents behaviors, the term *Agent* expresses only the "Type of Agents". The real separate Agents are the instances of this type. It is analogous with the terms Class and Object from the object oriented approaches. The real Agents are

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

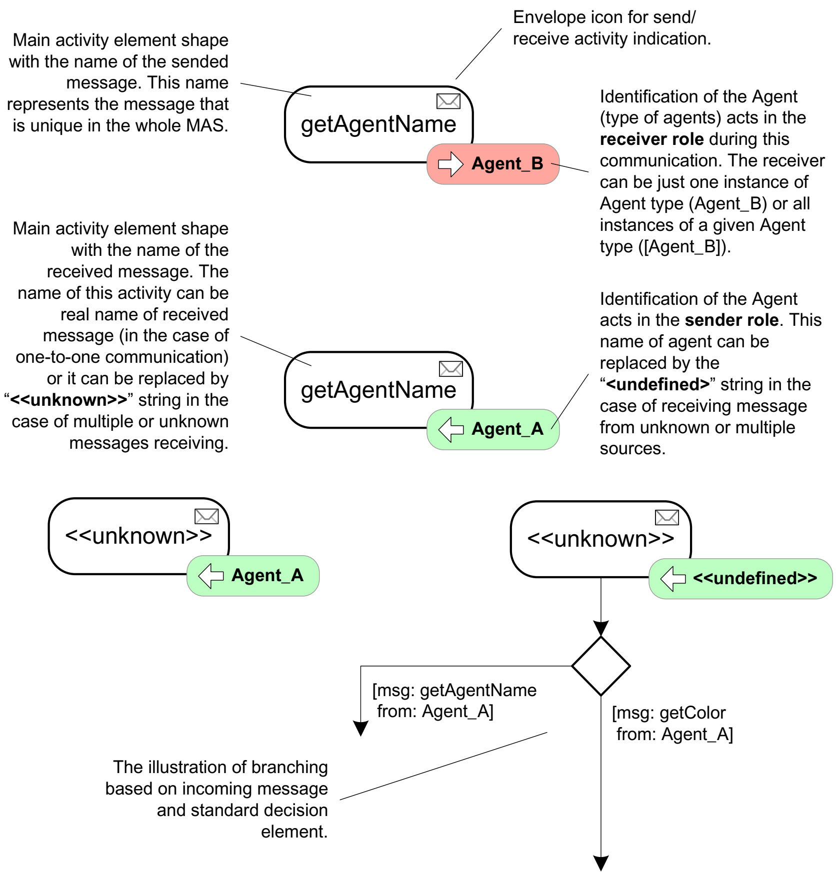

Figure 1: The example of new communication Activity stereotypes and illustration of decision point. These Send/Receive Activity nodes are able to be used beside the other standard Activity and Process nodes. Internal implementation of these new Activities is concerned only with preparation, en/decapsulation and receiving/sending of the message.

not issues of MAS design and modeling phase but they will appear in the implementation phase, simulation and operation of a given MAS.

UML ACTIVITY DIAGRAMS FOR MAS MODELING

The UML (Unified Modeling Language) is an essential tool for process modeling, both on the business level and analytic level of description (Vondrak 2004; Pilone 2005). It can be applicable for modeling of the internal behavior of Agents as well. The UML Activity Diagrams is a standard diagrammatic technique which describes the series of Activities, Processes and other control elements that together express the algorithm. They are especially suitable for modeling of agent behavior, though some modifications and extensions are required.

UML Activity Diagram Extensions

The *Agent Behavior Diagrams* could contain all the elements of the standard UML Activity Diagrams, and moreover some new elements can be used in the agent modeling process. These new elements are concerned with message passing among the Agents or with other specific attributes of MAS. In early phases of development, these extensions are supported by the implementation of special "Send/Receive Activities" which include an additional information about message content and message receiver/sender identification, see figure 1. The decision elements from standard Activity Diagrams are improved too. The modified "decision elements" and their output edges can hold some extra information. This information is usable for message-based determination of following control flow of the behavior led to the Agent's objectives. The new modeling or specification elements are the *Scenarios* of Processes. They are mentioned below.

The simple, clear and formal definition of the internal agent behavior is a precondition for moving to the next phases of the multi-agent software process (Radecky and Vondrak 2005). Thanks to the new stereotypes established in the *Agent Behavior Diagram*, it should be able to generate other types of diagrams (e.g. sequential diagrams, maps of the agents communication) as well as the source code templates of the Agents and MAS automatically (e.g. interfaces of Agents, class and method templates).

Internal Structure of the Agents

The agent internal behavior is specified by the algorithms expressed in the Processes. Each algorithm is modeled as just one *Agent Behavior Diagram* based on UML Activity Diagram. A couple of rules are bound together with creation process of these diagrams. At first, each Process, as well as diagram or algorithm, have to have just one "initial node" and just one "final node". This prerequisite is necessary for the further connection of the Processes together to model overall agent behavior. Also, the new term *Scenario* is introduced due to the existence of only one final node within the Process algorithm. Each Scenario describes one of possible finalizations of the Process and it combines the Output Objects related to this Process ending. The demand of the "well-formed" diagrams is also required. There is a set of general structural rules (e.g. the level of split/join nodes preservation, no crossing of the levels of control flows) that must be kept in mind during the drawing of diagrams (Aalst 1997). Thanks to these rules, the well applicable expressions of algorithms are obtained. It is able to verify them, to transform them to other forms or to process them by several formal tools.

Each Agent must have one *Primary Process* that covers whole behavior (life) of a given Agent. Each Process, as well as this Primary Process, can contain a control structures, Activities (atomic elements of algorithm) and references to other Processes. The usage of "Send/Receive Activities" is the only one way to connect all behaviors of separate Agents together. The Activities and Processes, except Primary Processes, can be specified globally within the MAS. These globally specified elements are possible to subsume into the behaviors of a set of Agents (Types of Agents). Some advanced features is concerned with the distribution of particular Process specifications. The Process with identical name can be defined within some Agent, as well as a member of set of Processes on the global level. In this case, some possible restrictions can be set up for each such Process. The Agent's local Process can extend a set of Realizations of this Process which is defined within the global repository or it can narrow this set only to the local Realizations owned by a given Agent. Some new notion is used in the previous sentences – *Realization*. The Realization is the modeling element that represents one of possible algorithms of Process firing. Each of such algorithms is expressed by one Activity Behavior Diagram.

METAMODEL OF OUR MAS MODEL

The figure 2 contains the UML Class Diagram that depicts a basic structure and elements of our MAS Model on the meta-modeling level. This abstraction is sufficient for better readability and clarity of MAS description from the structural and semi-formal point of view. The meta-model contains a definition of all MAS elements, e.g. *Activity*, *Process*, *Realization*, *Object* (Radecky and Vondrak 2005). The relations between elements and their restrictions are specified there too.

INTELLIGENCE WITHIN THE AGENT

Only the static and structural modeling of Agents is mentioned above, though, we want to speak about *Intelligent Agent* mainly. The Intelligent Agent is a standard Agent that disposes of certain kind of "brainpower". These capabilities are hidden inside the agent behavior and they can be founded in various points and situations of such behavior. The intelligence is ensured by application of several tools based on logic, artificial intelligence, etc.

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

- *Intelligence contained within the Activities* – the function of the logic is concerned with decision making and derivation of some new knowledge inside the Activity, e.g. weather forecast. This application of intelligence is hidden from the modeling perspective.

- *Intelligence of the control flows routing* – this application of several logic or intelligent tools is covered in the decision points – "intelligent decision point". The intelligent 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.

- *Intelligent selection of Process Realization* – the third task of intelligence subsuming into an Agent's life is concerned with the real-time running of MAS. Each Agent must try to realize its tasks and to solve the upcoming situations in order to meet its design objectives during its life. From this point of view, the standard Agent is consisted in finite and constricted description of its behavior already defined during the modeling phase of the Agent. Therefore, there is no way to change the behavior during the running of a given Agent. It is able to do this, in the case of *Intelligent Agent*.

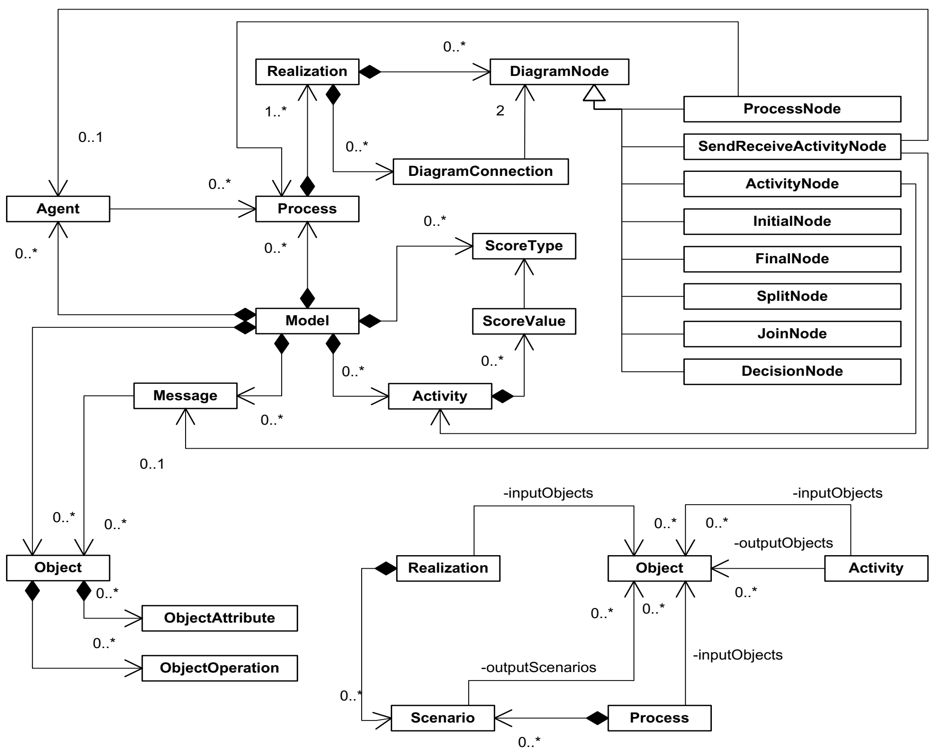

Figure 2: The meta-model of mentioned MAS Model approach

Intelligent Selection of Process Realization

The Intelligent Agent is based on the main life process approach that specifies just the framework of its behavior. The Agent can dynamically change the pieces of its own behavior according to the situations. This principle is denoted as *behavior reconfiguration approach*. The process of reconfiguration is founded on the replacement of a given part of the whole Agent's Process by another one that is the most suitable for current situation and conditions. A set of possible applicable algorithms, called *Realizations* (defined by one ABD), of each "reconfiguration point" (generally the Process node of Activity Diagrams) is defined for this purpose. These definitions of the Realizations can be distributed across the MAS. They can be saved in some global repository or within the particular Agents (some rules and restrictions have to be applied); actually they can be deliberated in the real-time too. The important and expected situation will appear whenever one reconfiguration point (Process) corresponds to the two or more Realizations. It is a time for *reconfiguration* of this Process.

The reconfiguration process and selection of the most suitable or applicable Realization can be implemented by a pure process or logic approaches. In this case, the each of them brings different advantages and disadvantages. The third possible implementation of it is based on the combination of both approaches. This perspective is mentioned in the next chapter.

The pure *Process Approach* is founded on the idea of the Realizations assignment to the parent Process already in the MAS modeling phase. This approach is a less flexible at the expense of almost zero error rates during the creation of a set of Realizations, in comparison with a logic approach. In other words, more effective Realization leading to a given Process objectives and that is not connected with such Process during the modeling phase may exists. Nevertheless, it is impossible to choose the Realization that indeed leads to the Process objectives but the execution of this Realization is mismatched, unusable or illegal.

In the case of the *Logic Approach*, the finding a set of suitable Realizations arises from the precondition that no connections between the Process and Realizations are there and each Process and each Realization have defined their Output Objects and objectives in accurate and clear form. The challenge of logic tool is to find (local or global repository can be taken into account with a respect to some restrictions or rules) a set of Realizations for the next steps of the reconfiguration algorithm. The description, properties, Objects and objectives of these Realizations have to be in accordance with objectives and Objects of a given Process. The logic approach when compared with process one is more flexible thanks to the fact that all relations between Process and Realizations are constructed automatically in the real-time just in the moment of request. Unfortunately, there is relatively hi-risk of mismatched, unusable or illegal Realizations selection. These problems are caused by demands for clear and accurate logic description.

Modified Reconfiguration Algorithm

The union of the process and logic approach within the reconfiguration process is described in following paragraphs. This method will be used in our researched Multi-Agents Systems. According to afore mentioned 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 the most suitable one.
   (a) the selection of applicable Realizations - based on Input Objects occurrence
   (b) the finding the most suitable Realizations - based on Input Objects values and properties, Scores, etc.
3. The Execution Phase - chosen Realization firing.

The combination of the process and logic approach in one algorithm seems to be more effective due to the ability to choose the best Realization. Simple process approach is not able to select the Realization in dependence on inner set of properties owned by Agents or Objects. On the other hand, the logic approach is not able to discard all Realizations which can not be executed at the time of Process firing.

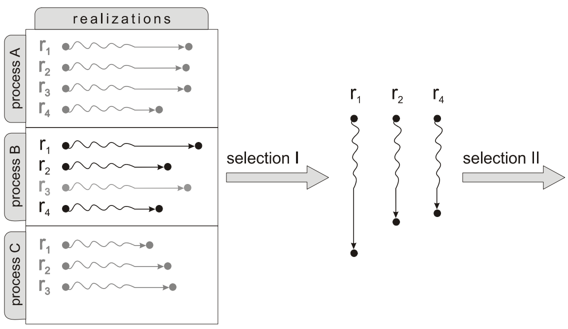

Figure 3: The basic scheme of selection phase method

The figure 3 shows the basic scheme of this reconfiguration algorithm. At the beginning, we already have a set of all Processes of our MAS and each Process has a set of related Realizations defined by the person that have modeled a given MAS. The number of Processes and their Realizations depends on the author of MAS Model and on the complexity of whole system. More Agents and their behaviors give rise to a higher number of Processes where each Process could be realized by different ways and it could be also described by many Realizations. The main idea of our method consists of two steps of selection which are described below.

Selection I (selection of applicable Realizations): This selection is based on the process approach again. First, the Process for reconfiguration making is selected (see Process B in the figure 3). It has to finish with a required state. The selected Process should be done by four Realizations (relations statically defined in the Model). However, some of the Realizations could not be executed sometimes because of many reasons. The Input and Output Objects and other conditions are defined in the Model as well as they come from the actual state of MAS. These features assigned to the Realizations are the groundwork of selection and filtering process. Therefore, the output of the Selection I is a subset of all possible Realizations of selected Process.

Each Realization of such Process can have other indicators like time, costs, etc. of its execution. Generally, these indicators could be defining as one value (cost) that is calculated based on some mathematical function. This solution is mathematically polite but it is not so user-friendly and predicative. Another solution is based on the individual values. These split indicators are generally called as *Score of Realization* and they are formed on the *Score Types* defined in the MAS Model. For the specification purpose, it is able to use a several quantities together with their units – boolean value (True/False), duration length (seconds, minutes, hours, days, years, etc.), distance (centimeters, meters, kilometers, etc.) or percentage values.
The Score of Realization is not defined during the MAS modeling but it is depended on the real realization algorithm and on the Scores of Activities within it. The summary Score of each Activity as well as Realization is given by the combination of all elementary Scores defined within the Activity (Realization). On the other hand, the summary Score of each Realization is calculated (in the moment of necessity) from the Scores of Activities occurred in the control flow of such algorithm of a given Realization.

Selection II (selection of the most suitable Realization): The second selection step is based on a logic approach. Till now, we have a set of Realizations which could be executed inside the MAS as a given Process. The simply question occurs: "Which is the best one?". The decision process is based on many parameters owned by Activities, Agents or Objects. The most important parameters are the values of the Scores of several Realizations and also the values and properties of Input Objects owned by this Realization. All of these parameters should be specified within the modeling phase of the MAS as well as they could be found in the time of reconfiguration. A predicate, fuzzy logic and Transparent Intensional Logic (TIL) or some alternative approaches, e.g. Formal Concept Analysis, could be used for the decision making during this step of the reconfiguration algorithm.

APPLICATION OF THIS

We would like to apply all of above mentioned ideas, approaches and results into the huge project that is concerned with the topic of "Intelligent Multi-Agent Systems". Within the scope of this research project, we have developed a couple of MASs based on this modeling approach and **Agent**Studio application, nowadays. Our testing MASs are focused on the problem of mobile Agents and their movement in the environment – some cooperation with GIS research group is realized. "Car Parking System", "Road Traffic Control System" could be an examples of them. The process specifications of the Models have formed the ground for whole development process of them. The example of some Process defined by the **Agent**Studio is depicted in figure 4.

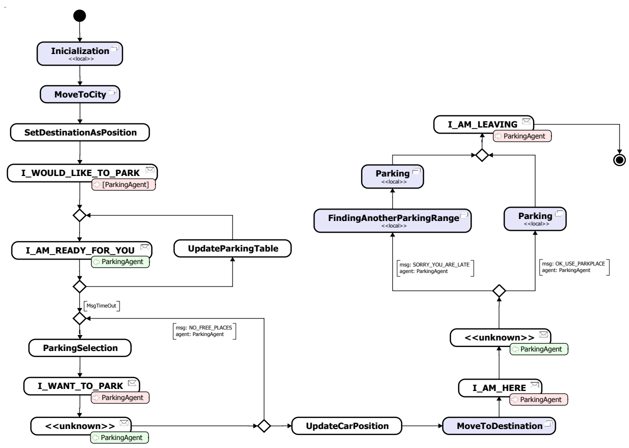

Figure 4: The example of Car Agent's Primary Process

FUTURE WORK

Although this research is in the initial phase, the basic ideas and some concrete conclusions and results concerned with the problematic of Intelligent Agents and their behavior modeling, as well as the approaches for reconfiguration of internal agents behaviors, are done and published at present. Now, it is necessary to define subtle rules, properties and facilities of all elements as well as the relationships between them and whole software process. These features are important for the following application of these ideas and approaches into the practice engagement. The development of software application that will provide methodology and application framework for modeling, controlling and operating of MAS system is an important goal of our research. The application **Agent**Studio is being developed for these purposes now. Thanks to this future work, it should be also possible to verify and simulate Agents as well as map the descriptions of Agents onto the source codes and real implementations of them. Above mentioned approaches and techniques produce the possibilities of distribution of knowledge and learning of the Agents, too.

CONCLUSION

This paper is concerned with the MAS technology whose future and power are, among others, dependent on the facilities and quality of the tools for its development process. This tools must be designed with a respect to the skills of normal users that will be a "modelers" of the MAS and that will determine the objectives, requirements and behavior of whole MAS from the real-world point of view. Such software application should be a one of the outputs of these researches. The modeling and intelligence approaches, graphical nodes, meta-model, some other solutions, etc. were mentioned in this paper. The future possibilities and advantages of MAS, which will be developed thanks to these approaches, are sketched in this paper too. They will be elaborated during the future work and research.

This research 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"

References

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

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

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

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

Radecky, M. and Vondrak, I. 2005. "Modeling of Processes". *WOFEX 2005*, Ostrava.

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

Pilone, D. and Pitman N. 2004. *"UML 2.0 in a Nutshell"*, O'Reilly Media, Sebastopol.