A DESCRIPTION METHOD FOR MULTI-AGENT SIMULATION MODEL UTILIZING TYPICAL ACTION PATTERNS OF AGENTS

Taiki Enomoto, Gou Hatakeyama, Masanori Akiyoshi and Norihisa Komoda
Osaka University, Yamadaoka 2-1, Suita, Osaka 565-0871, Japan

Keywords: Multi agent simulation, Description of events, Typical action patterns, Event flow.

Abstract: Recently, there are various proposals on tool for multi-agent simulation. However, in such simulation tools, analysts who do not have programming skill spend a lot of time to develop programs because notation of simulation models is not defined sufficiently and programming language is varied on tools. To solve this problem, a programming environment that defines the notation of simulation model based on graph representation is proposed. However, in this environment, we still need to write programs about a flow of event and contents of agents' action and effect. So, we propose a description method for multi-agent simulation model utilizing typical action patterns of agents. In this method, users write about designs of contents of event based on typical action patterns which are "interrogative (4W1H) and verbs", and designs of a flow of event. In this paper, we executed experiments that compare time needed for examinees to generate programs by a conventional method and our programming environment. Experimental result shows the time to generate programs by utilizing our programming environment less than that by utilizing a conventional one.

1 INTRODUCTION

Recently, multi-agent simulation (MAS) is expected to be effective for simulation of complex system such as biology, ecosystem, social system and economics(MacNealy, 1999)(Axelrod, 1995). MAS is a method of simulation that includes some autonomous agents which act in spontaneous manners like a human and make interactions each other. The purpose of MAS is to analyse phenomenon generated by interactions of agents. Agents make action based on rules and affect other agents.

MAS system consists of these local rules. There have been some studies of MAS, and various MAS tools have been developed (Swarm(Minar et al., 1996), Repast(North et al., 2005), artisoc(kke)) and used for many simulationDHowever, analysts spend a lot of time to describe a simulation model of the problem, because there is no way of understandable notation of simulation models based on feature of MAS. Moreover, when analysts write the program codes of simulation, they have to use specific programming language of MAS tool which makes them spend a lot of time to acquire programming skills.

To solve the above-mentioned problems, a programming environment that defines the notation of simulation model based on graph representation is proposed(Hatakeyama et al., 2007). However, in this environment, we still need to write program codes about a flow of event and contents of agents' action and effect.

In this paper, we propose a description method for multi-agent simulation model utilizing typical action patterns of agents. In this method, users write about contents of event based on typical action patterns, which are "interrogative (4W1H) and verbs", and designs of a flow of an event.

2 A MULTI-AGENT SIMULATION PROGRAMMING ENVIRONMENT

2.1 Outline of Programming Environment

Analysts who run simulation roughly sketch interactions of objects in simulation, and they design a simulation model by embody their understanding. Next,

Enomoto T., Hatakeyama G., Akiyoshi M. and Komoda N. (2008).
A DESCRIPTION METHOD FOR MULTI-AGENT SIMULATION MODEL UTILIZING TYPICAL ACTION PATTERNS OF AGENTS.
In *Proceedings of the Third International Conference on Software and Data Technologies - ISDM/ABF*, pages 316-319
DOI: 10.5220/0001900303160319

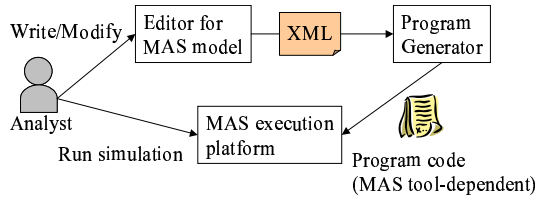

Figure 1: Outline of programming environment.

they write program codes based on the simulation model and run simulation with MAS tools. We can use several MAS tools as execution platforms, but these tools do not support to describe a simulation model and develop programs. To solve these problems, a programming environment that defines the notation of simulation model based on graph representation is proposed(Hatakeyama et al., 2007).

Figure 1 shows overall configuration of our simulation programming environment. First, analysts describe a simulation model of target problem by utilizing the editor for MAS model. Secondly, this simulation model is transformed into XML (eXtensible Markup Language) to represent information of the simulation model. This XML data is interpreted and transformed into a simulation program code by the program generater. We use "artisoc" as MAS execution platform. With MAS execution platform, we run the generated simulation program code and get simulation results.

2.2 MAS Model Editor

In our simulation programming environment, we need some elements to define the model of a target problem. Figure 2 shows these elements and a description of model based on these elements. In this example, the relationship between shops and consumers is shown in the upper part of Figure 2. First, we have to describe an agent itself and interaction of each "agent" such as consumers and shops. An agent has "attributes" which represent an agent's state. Also, an agent has "action" which influences other agents behavior or attributes. We need these 3 elements to describe a simulation model. Then the simulation model based on graph representation is depicted in the lower part of Figure 2. An agent's attribute is represented as "variable node", an agent's action is represented as "event" and an effect is represented as "arc".

An agent's attributes, action and effect are represented as a node, an event and an arc in MAS model respectively. However, contents of action and transitions of event are not represented well. So, we propose a description method for multi-agent simulation model utilizing typical action patterns of agents to support users to describe such events.

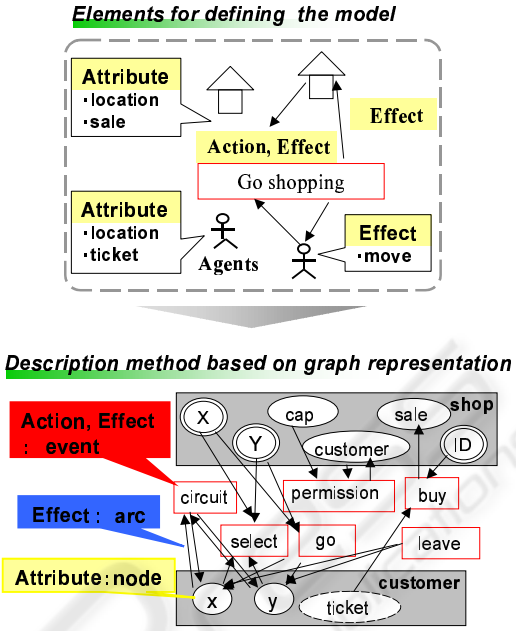

Figure 2: A description method of model.

3 EVENT DESCRIPTION

3.1 Outline of Event Description

Contents of events that agents behave independently are able to be described by utilizing "when" (when a condition is satisfied), "who" (an agent), "how" (following the rule), "what" (object) and "verb" (do). In this paper, we define "interrogative (4W1H) and verbs" as typical action patterns of agents and propose a description method for multi-agent simulation model utilizing typical action patterns of agents.

In addition to use of the interrogative (4W1H) and verbs for expressing the contents of agent events, we introduce "flow diagram of event transitions" for expressing interactions between agents.

3.2 Design of Event Contents

Figure 3 shows outline of design of event contents. We prepare pull-down choice lists to design event contents about "interrogative (4W1H) and verbs". In case of designing event that is selection of shop and if we want to design such as "if distance between a customer and a shop is within a certain value that we decide in advance, the customer selects the shop.", we describe a model such as "When => Always", "Who => customer", "How => Min (Distance (customer, shop))", "What => shop" and "Verb => Select One".

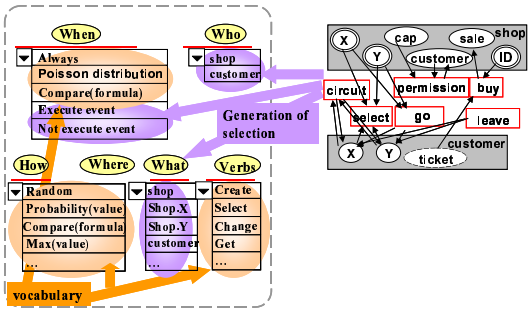

Figure 3: Design of event contents.

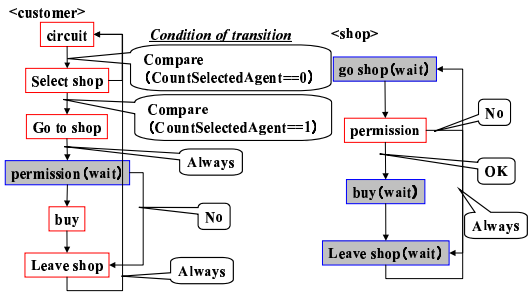

Figure 4: Example of a flow of an event.

There is a problem that it is difficult to prepare all verbs which are used whole entire world to design event contents. So, in this paper, we defined vocabulary of verbs by putting the verbs that we use everyday together through the lens of the agent world.

3.3 Design of a Flow of an Event

Figure 4 shows a flow of an event. The transition of an event is described about each agent in the flow of an event. The condition of transition is described in each arc by referring to the vocabulary that is stated in section 3.2. Also, timing of firing the event and the procedures that the event occurs are described in the flow of the event. In the Figure 4, the state "wait" represents that one agent waits for other agents to complete an event that it is necessary to be completed for one agent to transit next event.

4 AUTOMATIC GENERATION OF PROGRAMMING CODES

MAS model that is described by utilizing a model description method based on graph representation are represented in a form of XML. Event contents that are included in the MAS model and information of transition of events are represented in the same form. The XML file is input into the program generater and

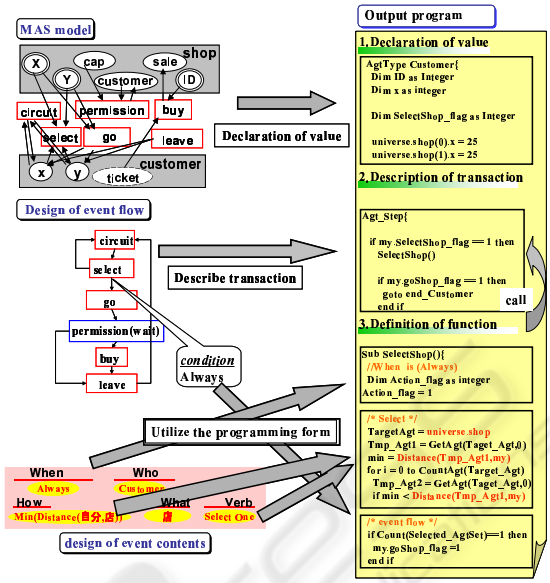

Figure 5: Outline of automatic generation of programming codes.

converted to program codes. Figure 5 shows the outline of automatic generation of programming codes. Also, we use "artisoc" as an execution platform in this paper. In "artisoc", description of agents is made up of "declaration of value that agents have", "agents rules that are executed per unit time" and "definition of function by users". In Figure 5, MAS model corresponds with declaration of value, design of event flow corresponds with agents rule and design of event contents corresponds with definition of function.

5 EVALUATION

We evaluate our description method by executing experiment on "shopping mall model".

5.1 Generation of a Shopping Mall Model

Figure 6 shows outline of a shopping mall.

- Customers arrive at the west gate or the south gate by some probability.
- Customers move in the shopping mall freely.
- Shop B passes coupons at the west gate and customers receive them at a constant rate.
- Customers go into the shop if distance between customer and a shop is within a constant value.
- Shops do not allow customers to go into the shop when each store capacity is not sufficient. And

customers flounce in the shopping mall freely again.

- If customers in shop A have a coupon, they leave the shop and do away with their coupon and they flounce in the shopping mall freely again. In other cases, customers buy goods.
- When customers finish buying goods, they leave the shopping mall
- The shops count the number of their customers

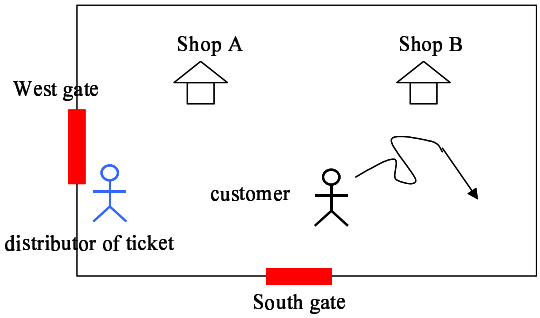

Figure 6: Outline of shopping mall.

5.2 Generation of Model by Examinees

5.2.1 Precondition of Experiment

First, examinees receive an explanation about the shopping mall model that is described in section 5.1. Secondly, they generate the shopping mall model by utilizing our event description method. Finally, we compare required time, which examinees generate a model by utilizing our approach and by writing raw program codes for "artisoc".

Two examinees have programming experience, however they inexperienced in MAS or "artisoc". So, they learn how to use "artisoc" by reading its manual.

5.2.2 Result of Experiment

Table 1 shows the result of experiment.

Table 1: Result of experiment.

	artisoc		Our approach		
examinee	Time (min)	Codes (line)	Time (min)	Pattern of action	Event flow
A	210	136	90	22	5
B	210	99	100	21	5

In experimental result, it is clarified that time, which is required to generate a simulation model by utilizing "artisoc", is reduced by utilizing our event description method. Additionally, our event description method is easy to use for not only person of experience but also amateur programmers of MAS.

However, simulation sometimes does not go well, because users draw the model in wrong manners. For example, in case of "leave shop event", users have to describe processing to reduce the number of customers. However, because of a perceived notion that the number of customers reduces naturally, users fail to describe this procedure explicitly. So, this simulation does not go well. We need to consider a method to bridge the gaps between users' perceived notions and codes.

6 CONCLUSIONS

In this paper, we proposed a description method for multi-agent simulation model utilizing typical action patterns of agents in a programming environment that defines the notation of simulation model based on graph representation.

In this paper, we executed experiment and compared required time, which examinees generate a model by utilizing our approach and by "artisoc". As a result, it is clarified that time is reduced by utilizing our event description method. Additionally, an event description method that we propose is easy to use for not only person of experience but also amateur programmers of MAS.

REFERENCES

Axelrod, R. (1995). The convergence and stability of cultures : Localconvergence and global plarization. Technical report, Working Paper 95-03-028,Santa Fe Institute.

Hatakeyama, G., Kimura, K., Akiyoshi, M., and Komoda, N. (2007). A programming environment for multi-agent simulation based on graph representation. In *Proc. of 2007 Summer Computer Simulation Conference (SCSC 2007)*, in CD-Rom.

kke. Mas community. http://mas.kke.co.jp/index.php.

MacNealy, M. S. (1999). *Strategies for Empirical Research in Writing*. Allyn Bacon.

Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The convergence and stability of cultures. Technical report, Working Paper 96-06-042,Santa Fe Institute.

North, M., Howe, T., Collier, N., and Vos, R. (2005). The repast simphony runtime system. In *Proc. of Agent 2005 Conference on Generative Social Processes*.