Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds.

FUSE: A MULTI-AGENT SIMULATION ENVIRONMENT

Kensuke Kuramoto

Graduate School of Industrial Technology
Nihon University
1-2-1 Izumi-cho, Narashino, Chiba, JAPAN

Masakazu Furuichi

College of Industrial Technology
Nihon University
1-2-1 Izumi-cho, Narashino, Chiba, JAPAN

ABSTRACT

We propose a new integrated multi-agent simulation environment "FUSE" which is designed for hierarchical organization behavior modeling. Recently, multi-agent simulations are getting more important in many fields such as planning of a large-scale disaster evacuation and military operations (Cil 2010, Nakajima 2008, Persons 2005). Since such simulations are characterized that they are composed of multiple of heterogeneous organizations, and the number of agents is very large, we have to provide an effective and efficient simulation environment. We have focused on a decision making model of organizations in the real world (Kuramoto 2012), and have implemented FUSE in Java. We have also proposed CaSPA which is based on a goal directed reasoning algorithm, and implemented reasoning rules in Java to show the basic functionality. Moreover, we have expanded foreign language interface to FUSE, and implemented CaSPA's reasoning rules in JRuby to show the applicability and expandability of CaSPA.

1 OVERVIEW OF FUSE

FUSE is designed for hierarchical organization behavior modeling shown in Fig. 1. As this figure shows, a hierarchically structured organization is composed of a boss and his subordinates who are super leader, leader or members. In the same organization, an objective is shared among all members. When an objective is given to a boss, then he breaks down the objective into sub-objectives and give them to his subordinates, and it is recursively done until the subordinates becomes empty.

FUSE is a set of libraries required to build large scale multi-agent simulation, and they are mostly written in Java. The developers can build their multi-agent simulation using FUSE, and they can build their agent's behavior rules in Java. However, since writing behavior rules in Java requires advanced programming skills, we have expanded FUSE so that script languages can be used for rules. Latest version of FUSE supports Java and JRuby for writing behavior rules, and the mechanism is extendable to other script languages.

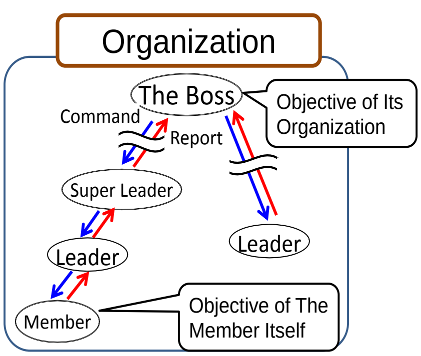

Fig. 1 Organization Model

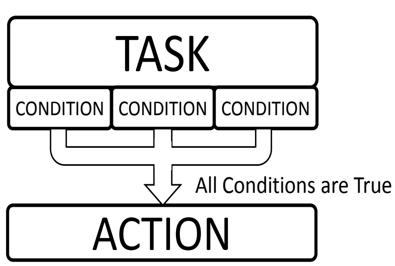

Fig. 2 Task, Condition and Action

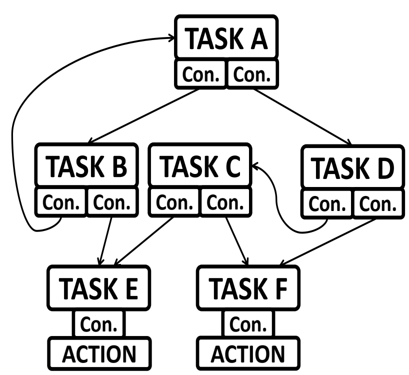

Fig. 3 An example of Task Graph

2 BEHAVIOR RULES AND CASPA

One of the most widely used agent's behavior rules is IF-Then rules. However, in a large scale complex multi-agent simulation such as a disaster evacuation or warfare between organization, the number of IF-Then rules becomes huge and it will get hard to maintain. In order to solve this problem, we have proposed a Cascadable Subgoals Production Algorithm (CaSPA) which is based on the idea of a goal directed reasoning algorithm.

All agents aim to achieve their goals, and the goals are derived from the objective of the organization given to the boss. In our algorithm, reasoning is achieved by three kind of objects, Task, Condition and Action (Fig. 2). Task defines the work to achieve objective, and also defines the cost required to do it. When all of the Conditions are True, the Task becomes possibible to execute, an Action is taken and the Task is completed. When one of the Conditions is False, reasoning mechanism of CaSPA tries to make that Condision to be True, then create new Tasks. When one of the Conditions is Unknown, reasoning mechanism of CaSPA tries to make that Condition to be Known (True or False). CaSPA proceeds the above process untill the generation of task graph (Fig. 3) is completed.

The reasoning process of CaSPA is controlled depending on the cost of the Tasks, and it is done by utilizing a task graph. In this figure, the leaves of the graph shows the executable tasks that are possible to do its Action. CaSPA executes these leaf tasks by order of minimum total cost.

3 EXAMPLES

We have implemented several examples to show the functionality and expandability of FUSE and CaSPA. Fig. 4 is a fire extinction at disaster simulation, and Fig.5 is a combat between two organizations. The objective given to the headquarter of fire brigade in Fig. 4 is “Keep safe all areas”, and it is “Overcome the enemies” for Fig. 5. Both worked as we've expected, and we will demonstrate them at the poster session.

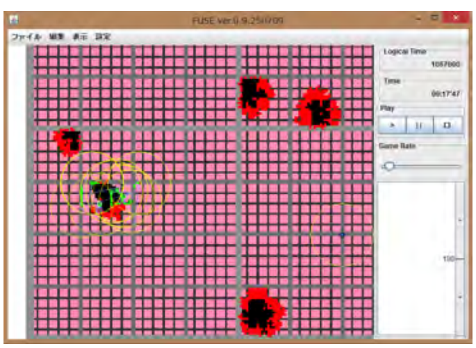

Fig. 4 Fire Extinction at Disaster Model

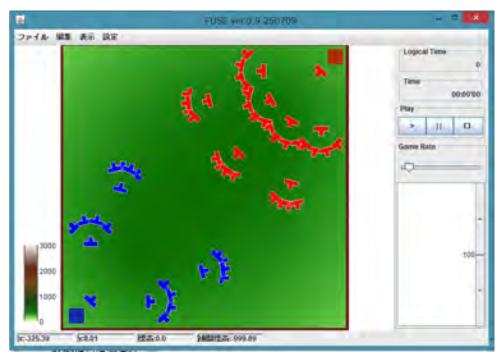

Fig. 5 Combat between Two Organization Model

REFERENCES

Cil, I. Mala, M. 2010."A multi-agent architecture for modelling and simulation of small military unit combat in asymmetric warfare", Expert Systems with Applications 37, pp.1331-1343

Kuramoto, K. Furuichi, M. 2012. "A Design and Preliminary Evaluation of Hierarchical Organizational Behavior Modeling Architecture", Proc. of JSST2012 International Conference on Simulation Technology

Nakajima, Y. Shiina, H. Hattori, H. Yamaki, H. Ishida, T. 2008."Augmentation of Experiment in Evacuation Navigation with Multiagent Simulation", IPSJ Transaction, Vol. 49 No. 6, pp.18 (2008)

Parsons, D. Surdu, J. Jordan, B. 2005. "OneSAF: A Next Generation Simulation Modeling the Contemporary Operating Environment", Proc. of Euro-Simulation Interoperability Workshop, 05E- SIW-19