A tutorial report for SENG 609.22
Agent Based Software Engineering
Course Instructor: Dr. Behrouz H. Far

Agent Based Simulation

A Report on: Modeling Multiagent Systems as Self-Organized Critical Systems

Carmen Zannier

1.0 Introduction

Simulation can be defined as, "an imitation or representation, as of potential situation or in experimental testing." It can also be defined as, "representation of the operation or features of one process or system through the use of another" [8]. Today, simulation is used for purposes ranging from work flow representation to military procedures. The question remains then - what techniques are best suited for varying simulation purposes? While object-oriented techniques are used for many types of simulation, also being explored is the use of agents in simulation. Specifically when simulating natural phenomena such as earthquakes and volcanoes, agent based simulation has proved to be successful. Traditional models have not been able to grasp the small physical processes that occur in nature which result in "catastrophes" such as volcano eruptions.

In studying such phenomena commonalities such as spatial and temporal scaling laws and slow and small driving disturbances were discovered, resulting in a belief by some that these natural phenomena are actually the result of small disturbances organized to result in chaos [1]. Consequently, the use of agents has been applied to simulation to represent these small disturbances and the resulting chaos. Three properties of agents play a key role in agent-based simulation. The first is an agent's ability to be autonomous and independent. The second is asynchronous message passing in agents. The third is social activity and reactivity [6]. Section two defines basic concepts needed to understand an agent based simulation architecture. Section three describes GEAMAS - Generic Architecture for Multi-Agent Simulation. Section four discusses some real world application and section five concludes the topic.

2.0 Definitions

"A **complex system** is inwardly driven by interactions between its components whose result exceeds the contributions of individual components" [2]. A **self-organized critical system** is a system wherein very small disturbances occur at a low level of the system, affecting the system little by little. When these small disturbances reach some threshold, the critical state is reached and the small disturbances aggregate to result in a catastrophe [1][4]. A typical example is that of a sand pile. Imagine placing grains of sand, one by one, on top of a sand heap. Each grain affects the pile by increasing the slope, and adding weight to the pile. At some point (or threshold) a grain is placed on the pile and an avalanche occurs. The critical state has been hit and a catastrophe results. Three major properties of SOCs are 1) their behaviour is non-linear and is governed by response to a threshold 2) the system experiences a slow driving force that induces small disturbances 3) descriptions of its behaviour exhibit scale invariance [1]. The small disturbances are **micro-behaviours** of the system and the chaos are **macro-behaviours**. The **emergence** of chaos is the appearance of new patterns that were not previously identified and are irreversible in the system [5]. Re-introducing properties of the emergence into the lower level small disturbances as constraints they must apply on their own structure is termed **back-propagation** [4]. As new patterns are generated dynamically from local interactions and the behaviour of underlying levels are determined by the new patterns, self-organization is a type of emergence. **Decomposition** is the transfer of information from higher levels to lower levels. **Recomposition** is the transfer of information from lower levels to higher levels. The following section will use these terms to detail a Generic Architecture for Multi-Agent Simulation.

3.0 GEAMAS

"The aim of the [GEAMAS] platform is to provide a complete toolkit as a virtual laboratory for designing a large scope of dynamic systems and providing generic interfaces to set and control the simulation" [4]. The GEAMAS structure consists of three dimensions: Knowledge Abstraction, Software Design, and Services. The Knowledge Abstraction Dimension manages the complexity of systems, the Software Design Dimension is like a library to develop simulation systems and the Services Dimension offers an environment to be extended [3]. The focus here is on the Knowledge Abstraction Dimension which consists of four layers. The Object layer contains all classes used or derived in upper layers. It is also termed the

language layer. The GEAMAS layer is the core of the platform and will be discussed in detail. The Application layer is the GEAMAS layer applied to a specific domain and the Simulation layer is an instance of the application. These layers all contain identical layers inside of them. A language layer describes Java as the language of choice for clarity and cleanness and the necessity to use applets. The agent layer uses agent capabilities such as autonomy and independence in conjunction with Java capabilities. Lastly the GUI layer hides complexity form the user [4].

3.1 Abstraction Levels

Where the GEAMAS layer meets the agent layer is where the meat for discussion resides. Three abstraction layers exist. A micro layer describes atomic and deterministic micro-behaviours performed by micro-agents. A macro layer looks at the system as a whole and observes and analyzes emerging macro-behaviour. It was initially believed that these two layers were sufficient but upon further analysis a medium layer was implemented in order to arrange intermediate structures.

3.1.1 The Agent Model Unit

The Agent Model Unit represents an agent as an atomic entity, or a micro-agent. Each micro-agent performs some task or behaviour without knowledge of global constraints or the domain and is thus termed reactive [2]. A micro-agent's behaviour may be internal, consisting of actions not involving external solicitations. A micro-agent's behaviour may also be external where a micro-agent is required to update it's state vector, compute a constraint to propagate to its neighbouring agents, chose the neighbouring agents to which it must propagate a message and lastly, actually propagate the message. The interaction protocol of the micro-agents is interesting to note. A micro-agent does not actually implement interactions itself, but instead is looked at as knowledge representation component only. Two mailboxes (*In* and *Out*) maintain the asynchronous nature of the micro-agents. The *In* mailbox contains all messages the should be performed by the micro-agent. These messages are ordered by intensity. The *Out* mailbox contains all messages a micro-agent must propagate to its neighbourhood after it has performed the message locally [4]. All micro-agents communicate with each other and use knowledge gained from other micro-agents to evolve and update their internal state [4]. "Casual links are dynamic conceptual links which define interactions between components in order to satisfy a common goal" [2]. An agent's state can be represented by a vector P, where the coordinates describe internal properties. Thresholds define the micro-agent's critical state by limiting a subset of the internal properties (a.k.a. state parameters). If a threshold is hit, the agent is deemed unstable. Neighbouring micro-agents are informed (recomposition) and a trigger mechanism searches for similarities among micro-agents. If two agents are found to be similar to one another they are grouped together, thus forming a class of similarity [2]. This aggregation is the process of self organization.

Each time the trigger mechanism detects similarity between two or more agents, a global meta function is activated to aggregate the similar micro-agents and compute the new medium agents' state vector. The medium level is designed before simulation begins but the medium level agents appear spontaneously and dynamically when self-organisation occurs. The medium agent then re-introduces its properties to underlying micro agents. This back-propagation enforces constraints on the micro agents to which they must conform [4]. When micro agents

transfer their knowledge upwards to the medium level, recomposition occurs. Information about an agent's stability may be transferred from micro to macro level. Decomposition occurs during back-propagation. Each reactive micro-agent is described by its micro-behaviours, its intentions and its evolution capabilities [6].

3.1.2 The Society Model Unit

The Society Model Unit is the macro level and describes the role of the system, the interface and the organization of the entire system. The macro level agents create and maintain all underlying agents. The system's role is a compilation of global events and global constraints, its interface is the required input output parameters needed for simulation and the organization is the structure of the inter-related agents [4]. The structure can be represented as a network, where the agents are nodes and their interactions are edges. Edges could even be given an intensity to show the strength of the relationship between agents. The intensity of an edge decreases as distance between two agents increases in the topological structure. The Society Model Unit consists of two mechanisms: Decomposition Mechanisms and Global Self-Organization Mechanisms.

Decomposition distributes a global event over the whole network so that lower level agents can perform the required tasks. Three functions must exist. First, a message must be sent to all low level micro agents. The message may be a constraint to which the micro agents must conform or a task that they must perform. The second function is one that "defines a process responsible for localizing the necessary agents to perform an event type that remains consistent with the goal of the society" [4]. Lastly an application-specific filtering function takes an external event as input and provides sub-events as output. The Global Self-Organisation Mechanisms create the medium agents and perform back-propagation. A two-level architecture is not sufficient as the micro level agents are too fine-grained and do not contain enough knowledge about the entire system and the macro level is too general and requires some determinism, which does not exist here [4].

3.2 Emergence Scenarios

The medium agent, responsible for creation and destruction of new emergent structures, must be aware of the different emergence scenarios that exist. The first scenario is when two agents cooperate and a brand new structure emerges. Here the new structure is encompassed by the medium agent and the two micro-agents adapt according to the back-propagated constraints. The second scenario is one where one of the two agents already belongs to an organizational structure. The medium agent must ensure the other agent is working towards the same goal and if so, must integrate the other agent. "In natural phenomena understanding, the set of goals is assumed to be the *global goal* or *global watching behaviour* and the own capabilities of the agent the *local goals*" [6]. In another scenario two agents belong to different medium agents. If the two medium agents are compatible (have the same goal), they can merge. A fourth scenario is where one micro agent ceases to participate in the common goal, then it must be removed. Lastly, if a micro-agent is the only agent in a structure and it disappears, the macro agent must kill the corresponding medium agent [5].

4.0 Application

One case study provided is the study of La Réunion Island, a hot spot on the earth's crust found in France. The focus is on the Piton de la Fournaise volcano. The main components of the volcano are: rocks from previous eruptions, lenses just under the main crater which form a superficial reservoir, dykes from cooled down lava and interconnected fractures through which magma circulates during eruptions. SOC behaviour occurs anywhere from a few months time for shallow induced earthquakes to $10\text{-}10^3$ years for eruptions [1]. The circulation of magma was the main focus for micro-behaviours. Macro-behaviours were the result of behaviours emerging from micro behaviours, as previously stated.

The micro-agents defined are rocks and magma lenses. The lens agents use the rock agents to exchange magma. Rock agents direct the exchange while lens agents determine the volume. A lens agent's critical state is related to its internal pressure. When the threshold is reached, excess magma is transferred to other lens agents. "Internal pressure can increase and exceed its local critical threshold either when the lens agent receives some fluid from other agents or from external injections, or as the result of an internal autonomous process, namely crystallization" [1]. Aggregated rock and lens micro-agents model whole lens and rock sections, which contain similar properties. The macro-behaviours relating to volcano simulation are frequency and volume of eruptions or lens expansion and pressure variations, and finally, dyke formation or intrusion. Eruptions are related to specific lens micro-agents and a macro-agent possesses all data related to the micro-agents involved in the eruption. Macro-agents also distribute pressure over lens and rock micro-agents, ensures physical laws are globally enforced and controls external magma injections into lens micro-agents [1]. As a result the 1920-1992 Piton de La Fournaise eruption patterns were recovered.

A second application to Agent-based simulation is found in the air force. While this simulation does not necessarily follow the previous architecture, it gives promise to the use of agents in simulation. The Agent-based Modeling and Behavioural Representation program in the Human Effectiveness Directorate are attempting to use agents to simulate command post exercises for the United States Air Force [7]. Currently, some simulation systems exist but the behaviour is not entirely realistic. Specifically the ABMR is focusing on information operations in the hopes of simulating the execution of information warfare to influence the enemy's ability to make decisions. An example of some of ABMR's work is Raytheon, a "role-playing intelligent controller node which will be an autonomous agent-based model capable of "seeing" changes to the battlefield and reporting them back to the role-player" [7]. ABMR intends to compare various modelling approaches in the hopes of attaining various new approaches to simulation.

5.0 Conclusion

Simulation has the potential to save millions of dollars by testing an idea without actually implementing it. It also has the potential to clarify rather vague understandings of natural phenomena. Traditional methods have been insufficient as they are based on linear concepts. In using agents to simulate natural phenomena, local phenomena can be represented as well as emergent phenomena which arise through the interactions of local entities. A two-level approach was attempted and deemed insufficient as the issue of emergence must be dealt with on its own

level. Consequently a three-level approach has been presented. A micro-level concerns atomic activities. A medium-level concerns the aggregation of these atomic activities due to a critical state being reached. Finally, a macro-level concerns the overall maintenance and organization of underlying agents in a society. Messages are passed up and down a topological structure affecting various agents along the way. The activities of various micro-agents lead to the emergence of new medium agents containing irreversible properties that are passed back down to micro-agents who must conform to these new properties. The emergence represents some global activity termed a catastrophe. All of this activity occurs in the GEAMAS layer of a complex system. It has proven successful in describing past activity of volcano Piton de La Fournaise in France.

References

[1] Giroux S.; Marcenac P.; Quinqueton J.; Grasso J.; "Modelling and Simulating Self-Organized Critical Systems" Centre of Research LICEF / UER Science and Technology Montreal, Que., Canada,

[2] P. Marcenac, "Modeling MultiAgent Systems as Self-Organized Critical Systems", 31st Hawaii International Conference on System Sciences, HICSS-31, IEEE Computer Society Press, Vol. 5, Dan Dolk eds, Pages 86-95, 1998.

[3] Courdier R.; Marcenac P.; Calderoni S.; "Zooming on a Multiagent Simulation System: from the Coneptual Architecture to the Interaction Protocol", IREMIA, University of La Réunion

[4] Marcenac, P.; Courdier R.; Calderoni S.; Soulie J.; "Towards an Emergence Machine for Complex Systems Simulations", Springer-Verlag Berlin Heidelberg 1998.

[5] Calderoni S., Marcenac P., "Emergence of Earthquakes by Multi-agent Simulation" in Proceedings of European Simulation Multi-Conference, ESM'97, Istanbul, Turkey, SCS Int. Publishers, pp. 665-669, 1997.

[6] S. Giroux, P. Marcenac, S. Calderoni, D. Grosser and J.R. Grasso, "A Report of a Case Study with Agents in Simulation", First International Conference on Practical Applications of Intelligent Agents and MultiAgents Technology, London, UK, April 1996.

[7] Young M.; "Agent-Based Modeling and Behavioural Representation", http://www.afrlhorizons.com/Briefs/0006/HE0009.html

[8] http://www.dictionary.com/cgi-bin/dict.pl?term=simulation