

# **FFCAS: A FLEXIBLE AGENT-BASED SIMULATION FRAMEWORK FOR COMPLEX ADAPTIVE SYSTEMS**

Ji Wu, Chaoqun Ye, Shiyao Jin

*Collage of Computer Science and Technology,*

*National University of Defense Technology, 410073 Changsha, China*

*{wwuji, cqyie, Syjin1937}@163.com*

**Abstract:** Agent-based simulation methodology has shown merits to handle complex adaptive systems' spontaneity, non-linearity, uncertainty, relevancy and emergence. Based on the agent theory of modeling and simulation, this paper proposes a flexible, scalable and cross-domain framework for simulating complex adaptive systems. After discussion on the design of FFCAS, applications on FFCAS in different complex domains, and empirical results are presented as experimental validation.

**Key words:** multi-agent simulation, complex adaptive system, ecology, public attitude

## **1. INTRODUCTION**

Complex adaptive systems (CASs) have been identified everywhere both in the nature and human world. For CASs' spontaneity, non-linearity, uncertainty, relevancy and emergence, it is hard to model and analysis complex system with mathematical and other formalized methods. Agent-based modeling and simulation methodology has shown good prospect in dealing with the CASs difficulties. Existing agent-based simulation applications can be seen in domains include computational economic[1], ecology[2], military strategy[3], and certain social science specific as our examples in public attitude etc.

So far, many tools have been developed to facilitate the modeling and developing for the CAS simulation application, such as popular SFI Swarm[4] which was implemented in Object C, Cormas[5] in Smalltalk, and RePast[6] in Java. Unfortunately, none of these testbeds meets in large-scale CAS problem. Thus, flexible, scalable agent-based framework for CAS simulation is in urge.

FFCAS (Flexible simulation Framework for Complex Adaptive System) is a Java-based CAS simulation, integration, and development framework. FFCAS currently runs on cluster of workstations (COW) environments. The FFCAS design gives special attention to perfect current agent-based conceptual model and simulating very large communities of agents. It exhibits a significant degree of flexibility in dealing with multi-domain problems, scalability in permitting simulation for more than 4,000 agents.

The rest of this paper is organized as follows. To facilitate our discussion, section 2 outlines the agent-based CAS model embedded in the framework. Section 3 introduces design considerations and architecture of the framework. In section 4, applications built on the framework are presented as validate demonstration. Section 5 describes recent related work and concludes this paper.

## 2. ENHANCED AGENT-BASED CASS MODEL

Based on traditional theory of agent-based modeling and simulation and empirical researches, we regard that a complex adaptive system can be defined as such a tuple:

$$CAS_{Model} = \langle Environment, Agents, Micro Rules, Macro Rules, External Stimulations \rangle. \quad (1)$$

Here *Environment* denotes set of system situations; *Agents* represent system entities which exhibit autonomy, adaptability and sociability; *Micro Rules* denote agents evolvement patterns, restraints of interactions between agents; *Macro Rules* denote evolvement and constraints on system macro levels; *External Stimulations* denote external effects which destroy system's original balance and reaches a new balanced state. There is no difference from traditional models except *Macro rules*. Most of traditional models only keen on the micro level regulation. They ignore system level regulation effects on system. In fact, evolutions of complex adaptive systems depend surely on the hybrid effects of micro and macro regulation.

Proofs can be found in typical CASSs. Like Grassland-Ecological System[7], which describes the dynamic balance of numbers of species. Seldom traditional research considers the macro effects on the ecological system, like climate, grassland's or animals' growth cycles, and other

specie's aggression. Another case is theory of plate tectonics. Plate motions and collisions can be modeled in agent theory: plate is modeled as agent group while each agent represents unit area plate, and each agent's properties include altitude, crust stress, tenacity etc. When see into the continent plates' collision, agents at the plate boundaries and fragile zones will tune their properties to exhibit the collision consequence such as altitudes ascend or descend; agent groups at fragile parts may depart from the plates to form new plates or islands.

Figure 1 demonstrates notion of the macro rules involved agent based CAS model. In phase I, target CAS was run with primitive agents, which evolved according to micro rules and external stimulations. Target CAS transfer from Phase I to Phase II, part of agents in the environment evolves to new types agents; and by effects of macro rules, environment at macro level also evolve. Namely, target CAS evolves according to hybrid effects of each agent's evolution and macro level evolution. In phase III, agents and environment advance continuously, and for the reason that primitive agents can not adapt to the new environment, they will eliminate to the end; on the contrary, evolutionary agent will thrive for their adaptabilities to the evolved environment.

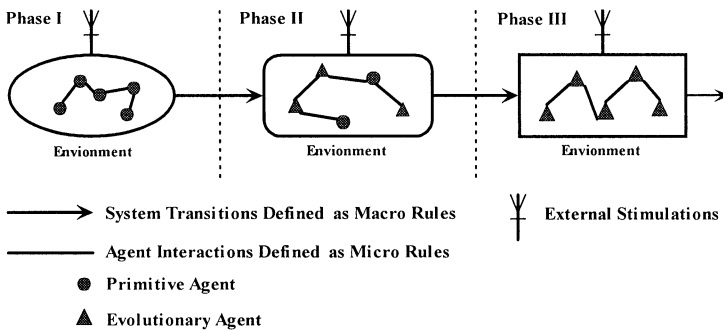


Figure 1. The Enhanced Macro Rules Involved Agent-Based CAS Model

### 3. FFCAS: AGENT-BASED SIMULATION FRAMEWORK FOR CAS

As discussed in section 1 and section 2, we can identify that requirements of CAS simulation platform may include two folds. One is perfecting the kernel model of CAS embedded in the platform and the other is matching growing scale of simulation computation. Here, when presents design

criteria of FFCAS, we mainly keen on two aspects: one is specifications initiated from the above macro rules involved agent-based CAS simulation model; the other is on process efficiency and computation decentralizing. We call these two design starts are model-driven and computation-driven design criteria.

### 3.1 Model-Driven Design

UML class diagram is used to depict the FFCAS design derived from the above conceptual model. All elements of the agent-based CAS model in tuple (1) are exhibited and abstracted to classes. These classes and their connections are depicted as Figure 2.

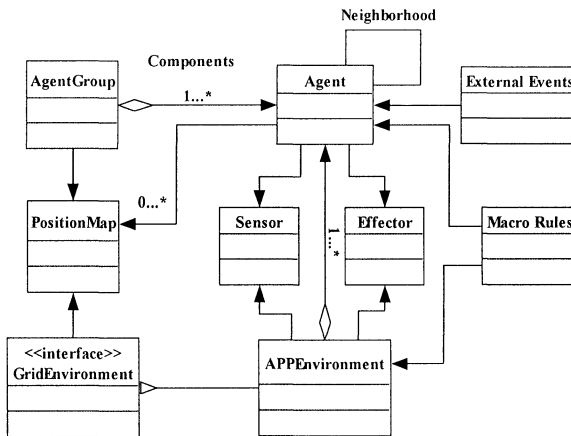


Figure 2. Class Diagram for the Framework

- **Agent** class is the basic element of the CAS. Its properties specify simulation related social and natural features; its methods generally implement agent's functions of environment data processing, decision, and motion. Connections among agents specified interactions and social relationship like neighborhood among them.
- **Agent group** class is a container for all agents. The class is constructed in purpose of providing a schedule console for all agents' activities.
- **GridEnvironment** class and **APPEnvironment** class implement CAS environment. And **GridEnvironment** class is constructed for exchange data between environment model and the display class namely **PositionMap**.
- **PositionMap** class is in charge of collecting statistics from agents and environment then display running information in a grid map.
- **Sensor** class and **Effector** class implement agent's sensor and action.

- **External events** class implements parameter variation on agents' properties resulted from external stimulations.
- **Macro Rules** class implements parameter variation both agents' properties and environment's parameters resulted from macro rules.

### 3.2 Computation-Driven Design

Java was chosen for FFCAS implementation for its excellent multi-threads techniques. As had been discussed in lecture, when Java program constructed as multi-thread mode, the program will execute in a parallel mode, and memory management and computation resource will be disposed at once a thread end. While other OO languages implemented program may run in a serial mode and instead of thread, process is used to run a program instance, which is memory consumed and uncontrollable. Other advantages of Java lie in its capabilities in modularization, platform-independence and openness.

Figure.3 presents the FFCAS's distributed computational architecture on 2 workstations. Each *agent* run as a thread and all these threads are put in the *agent hash table*. Schedule on agent threads completes in the *agent group* thread loop. Agent group, *environment* and *position map* are implemented as parallel threads. Agent threads and environment thread can be partitioned into multiple computation nodes. And synchronization and time management of three main threads, i.e., agent group, environment, position map, are maintained by simulation engine.

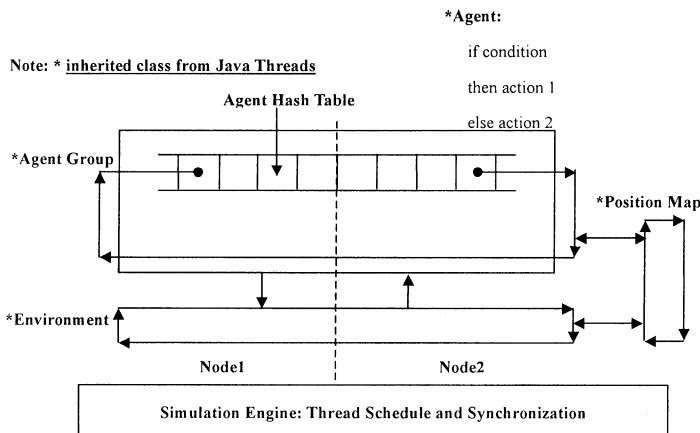


Figure 3. Distributed Computational Architecture of FFCAS

With this design framework, FFCAS has flexibility in model general CASS, for only common features were abstracted to the framework.

Secondly, multi-threads implemented framework ensures computing efficiency, no matter the domain applications run on one machine or cluster. Thirdly, scalability is acquired for the partitions of the agent hash table and environment model, and display and synchronization are detached running so they wouldn't bother the computation decentralized.

#### 4. CASE VALIDATION

Applications in different domains have been implemented on FFCAS for validation. Inter-domain flexibility, reusability and scalability of FFCAS is approximately measured in Table 1. Column 1 lists two different domains of FFCAS's application — Grassland Ecology System (GES), Pubic Attitude System (PAS) — thus providing some evidence of FFCAS's generality. Column 2 measures percent reuses of FFCAS codes across domain. There is 40% reuse in GES from the initial framework. While in building up PAS, reuse was improved to 60% for agent interacting in neighborhood, stimulation effects had been encoding and work was saving. Column 3 list agent numbers and Column 4 list the Grid Size in the two applications. In PAS, more than 4,000 agents and grid with size of  $150 \times 100$  are implemented. GES was run on 2 computing nodes and PAS on 4.

*Table 1.* Flexibility, reusability, and scalability data of FFCAS

Domain	Reuse	Agent Number	Grid Size	Computing Node
GES	40%	328	$64 \times 64$	2
PAS	60%	4192	$150 \times 100$	4

Grassland Ecological System (GES) simulate the dynamic balance on numbers of wolf—rabbit—grass pane. Cormas has implemented it as platform demo named ECEC[5]. We extended this work for comparative purpose. Extended GES simulations were run with about 328 agents (200 rabbits, and 128 wolves are initiated) and 820 grass pane, on a  $64 \times 64$ -cell grid. A COW of two nodes is used to support the GES running. Each type of agent runs on their strategy, i.e., grass panes are eliminated by rabbits and regenerate in their growth law. And rabbits survive on grass, and stay in any state of stuff, hungry and dead. Wolves survive on hunting rabbits, stay in any one of the same three states as rabbits. When in stuff state, animals jog on the grid in random manner; in hungry state, they search out for food, if not access food, their energy index continues decreasing. When energy index lower than zero, animal starves and is eliminated from the grid.

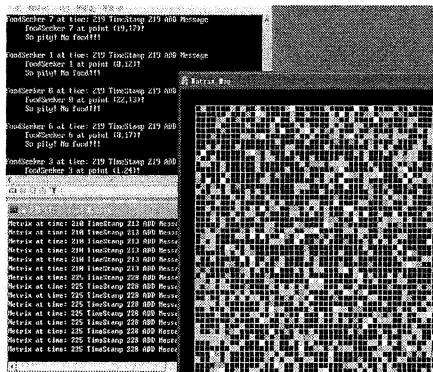


Figure. 4 Simulation scenario of FFCAS applied to Grassland Ecological System. Rabbits (red cells) stroll about in the grid and feed on grass pane (yellow cell), wolves (white cell) pursue rabbits, and environment will control the regeneration rules of grass pane according to the grassland growth law, climate. Two command windows list agent interaction messages between two distributed nodes.

The second example is Public Attitude System (PAS), as in Figure.5. Public attitude denotes the mass thinking and tendency on certain public event, such as approve, dispose or neutral. Empirical research works show that public attitude on certain event on each citizen's faith and faith environment around him/her. And faith immigration, namely, attitude change on the event, will happen when there are positive or negative feedbacks on the event. And final degree of the effects depend on citizen's social properties include age, gender, occupation, economic status.

PAS simulation was run with about 4192 agents (about 1048 approval agents, 838 disapproval agents, and 1886 neutral agents are initiated) on a  $150 \times 100$ -cell grid. A COW of four nodes is used to support the PAS running. The running strategy is that, i.e., agents each with certain belief were initiated using poll data, each of them interacted with their neighbors and belief was interfered by the environment around him/her and slowly tuned their own believes to the mean level. When a external stimulation which maybe a positive or negative feedback was scheduled, agents' believes transfer, and to what effect extent depends on agents' social properties such as age, gender, social status etc.

We choose the Grassland Ecological System and Public Attitude System as demonstration of FFCAS, for GES was regarded as member of the first generation multi-agent model, while PAS represents novel domain of agent-based simulation — social process simulation. In our developed cases, classical agent-based simulation application is extended in agents scale, more complex heuristic and enhanced model description relevant to the

enhanced agent-based CAS model. While PAS shows FFCAS's flexibility and scalability in supporting large scale social process simulation.

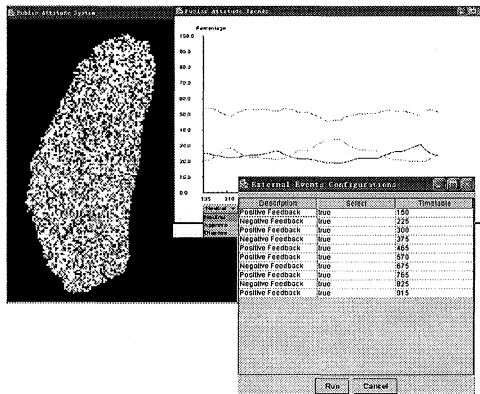


Figure. 5 Simulation scenario of FFCAS applied to Public Attitude System. Approval citizens (blue cells), Disposal citizens (green cells) and Neutral citizens (white cells) are initiated in the spatially-explicit environment. Citizen with certain belief tune their belief value according to its neighborhood and external feedbacks which are listed in the external event panel. In curve panel, number variations of three types of agents are demonstrated in real time.

## 5. RELATED WORK AND CONCLUSION

Many tools have been developed to facilitate the modeling and development for the CAS simulation app, such as popular SFI Swarm, Cormas and RePast. They implemented abundant tools and facilities for CAS simulation. For example, Cormas contains flexible agent and environment modeling facilities, display tools that include graphs and movies, and flexible data capture, and a CAS application can be built with small amount of encoding. However, these single processor depending platforms can not meet the rapid growth of scale of CAS simulation.

Other endeavors to improve the scalability of CAS platform include: building multi-agent simulation on the HLA/RTI[10], JADE[11], and other simulation middleware. However, each of them has to be improved before they can afford simulating a scalable CAS application. We know that HLA/RTI is not specifically agent-based, and agent-oriented services (message passing, agent communication languages, middleware services, addressing, coordination infrastructures, etc.) are not intrinsic part of HLA/RTI. JADE is a specific platform for simulating communications between small amount agents, it specifies none concept of environment, evolve rules which are kernel of CAS simulation. Despite their shortages, we consider that in future CAS simulation work, high performance



communication component and excellent services middleware will improve performance of current existing platform.

FFCAS is a Java-based CAS simulation, integration, and development framework. FFCAS currently runs on single workstation, and in cluster of workstations (COW) environments. The FFCAS design gives special attention to perfecting current agent-based conceptual model and simulating very large communities of agents. It exhibits a significant degree of flexibility in dealing with inter-domain problems, scalability in permitting simulation for more than 4,000 agents.

This paper presents FFCAS design criteria and our approach to a number of critical tradeoff. FFCAS shows novelty in breadth, flexibility and scalability, implements a enhanced macro rules involved agent-based CAS model and finally give results from the use of FFCAS in real experiments in realistic CAS domains. The framework offers a bridge for theory integration where gaps and contradictions become evident.

## REFERENCES

1. <http://www.econ.iastate.edu/tesfatsi/> (2005)
2. Craig, W., R.(ed.): *Flocks, Herds, and Schools: A Distributed Behavioral Model*. Computer Graphics, Vol.21, (1987) 25-34
3. Michael, B., Adam E.(ed): *CROCADILE-An Open, Extensible Agent-based Distillation Engine*. Information & Security, Vol.8, (2002) 17-51
4. <http://www.swarm.org> (2005)
5. <http://cormas.cirad.fr/> (2005)
6. Nick C. RePast: the REsursive Porous Agent Simualtion Toolkit.  
<http://repast.sourceforge.net> (2005)
7. Pepper, J.W., Smuts B.B.: *The Evolution of Cooperation in An Ecological Context: An Agent-Based Model*. Dynamics of Human and Primate Societies: Agent-Based Modeling of Social and Spatial Processes, Oxford University Press, Oxford(2000) 45-76
8. Wooldridge M, Jennings N R, Kinny D.: *The Gaia Methodology for Agent-Oriented Analysis and Designs*. Journal of Autonomous Agents and Multi-Agent Systems. Vol.3, (2000) 285-312
9. Tambe, M.: *Towards Flexible Teamwork*. Journal of Artificial Intelligence Research, Vo.7, (1997) 83-124
10. Wang, F., Turner, S., J., Wang, L.: *Integrating Agents into HLA-Based Distributed Virtual Environments*, The 4th Workshop on Agent-Based Simulation, (2003) 9-14