

 UNIVERSITY OF CALGARY	Course Number: SENG 609.22	Course Name: Agent-based Software Engineering
	Session: Fall, 2003	Department: Electrical and Computer Engineering
		Document Type: Tutorial Report

## **Multi-Agent Simulation Using SWARM**

**Ibrahim Jadalowen**

*(jadalow@cpsc.ucalgary.ca)*

Submitted as partial fulfillment of requirements for SENG  
609.22 (Agent-based Software Engineering)

Course Instructor: Dr. Behrouz H. Far

University of Calgary, Canada  
December 8<sup>th</sup>, 2003

## **Abstract**

The field of social simulation and agent base simulation modelling techniques is growing. With it grows the need for simulation tools. A way to reduce the effort of software development is to utilize generalized agent development tools. Today there exist tools that are based on the object oriented paradigm that provide libraries and graphical wizards.

In this paper we introduce the Swarm tool. It was developed at the Santa Fe Institute. It is an agent simulation tool designed to help researchers in building multi-agent simulation models. It also offers a useful graphical representation of the simulation results. A brief overview of the steps to build agent based models is also represented.

## **1. Introduction**

Simulation is a widely acknowledged method in different industries and research areas. The agent based simulation world is a wide unexplored world. Only few tools managed to survive. Swarm is one of those high calibre tools.

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 between agents. The third is social activity [5].

Memory management and time management in multi-agent simulation is an issue tackled by Swarm from a high level view. Swarm is based on the object oriented paradigm. It can be useful in simulating chaotic sequences, such as chaotic price sequences and the interactions consumers and vendors. It can also be useful in simulating natural phenomena such as earthquakes and volcanoes.

## **2. Swarm Overview**

Swarm is a software package for multi-agent simulation of complex systems. Swarm is intended to be a useful tool for researchers in a variety of disciplines. The basic architecture of Swarm is the simulation of collections of concurrently interacting agents: with this architecture, we can implement a large variety of agent based models. Swarm is a free software, released under the [GNU General Public License \(GPL\)](#). [2]

The primary feature of Swarm is the virtual machine. The virtual machine allows the researcher to describe agent behaviors one by one, agent by agent, context by context, all while keeping an exact notion of time and concurrency in the world. Swarm also makes it possible to compose or decompose hierarchies of agents.

Following the Swarm documentation (Swarm home page), we introduce a general sketch about how one might implement an experiment in the agent based modeling field. An idealized experiment requires, first, the definition of: (i) the computer based experimental procedure and (ii) the software implementation of the problem. [3]

The first step is translating the real world system (the physical system) of our problem into a set of agents and events. From a computer science point of view, agents become objects and events become steps in our program loops.

Using Swarm, We can avoid using the low level programming languages like C to manage both the agent memory structures and time steps. Swarm dismisses the memory management and the time simulation problems. A high level tool like Swarm still treats the events as objects, scheduling them in time-sensitive widgets (such as action-groups).[3]

### **3. Building Agent Models**

The idea of agents based systems can be described as a combination of Agent Based Modeling (ABM) and Intelligent Agent (IA) techniques.

In the Swarm context, we use the Object-Oriented Programming language Objective-C. According to the Swarm documentation, computation in a Swarm application takes place by instructing objects to send messages to each other. The basic message syntax is

```
[targetObject message Arg1: var1 Arg2: var2]
```

where targetObject is the recipient of the message, messageArg1:Arg2: is the message to send to that object, and var1 and var2 are arguments to pass along with the message.

According to the Swarm documents [1]:

Objective C messages are keyword/value oriented, that is why the message name message Arg1: Arg2: is interspersed with the arguments. The idea of Swarm is to provide an execution context within which a large number of objects can "live their lives" and interact with one another in a distributed, concurrent manner.

In the context of the Swarm simulation system, the generic outline of an experimental procedure takes the following form.

- i. Create an artificial universe replete with space, time, and objects that can be located, within reason, to certain "points" in the overall structure of space and time within the universe, and allow these objects to determine their own behavior according to their own rules and internal state in concert with sampling the state of the world, usually only sparsely.

- ii. Create a number of objects which will serve to observe, record, and analyze data produced by the behavior of the objects in the artificial universe implemented in step i.

- iii. Run the universe, moving both the simulation and observation objects forward in time under some explicit model of concurrency.

iv. Interact with the experiment via the data produced by the instrumentation objects to perform a series of controlled experimental runs of the system.

### **3. Interpreting the results**

Swarm has a tool capable of dealing with the representation of the results and, most of all, specialized in object-agents, an object - in the Objective C sense - being a piece of program which understands messages and reacts to them. Only if we are using a high level structured programming tool, it is possible to publish simulation results in a useful way.[3]

The fact that the Swarm structure has two different levels is very useful. There is the model level (and we can have nested models of models, or swarms of swarms) and the observer level which considers the model (or the nested models) as a unique object to interact with, in order to obtain the results and to send them to various display tools and widgets.[1]

### **4. Other Agent Simulation tools**

JAFMAS provides a framework to guide the coherent development of multi-agent systems along with a set of classes for agent deployment in Java. The framework is intended to help beginning and expert developers structure their ideas into concrete agent applications.[4]

### **5. Conclusion**

Sometimes it is difficult to understand and predict occurrence of events in indeterministic environments. Simulation can save millions of dollars and much effort by running a computerized model without actually implementing it. Agent-based simulation is very useful in clarifying vague understanding of economic, social interactions and chaotic sequences.

Swarm is a very useful high level development tool for building agent based simulation models. Swarm is a set of libraries that facilitate implementation of agent-based models. It is based on the object oriented paradigm, where agents are represented by objects in a time sensitive environment. Each agent is treated as a separate autonomous entity.

The development of agent based simulation is still at its infancy when compared to other Software Engineering disciplines. Much research and development room is available.

## References:

[1] Swarm Development Group: <http://www.swarm.org>

[2] MINAR, M., Burkhart, R., Langton, C., Askenazy, M. 1996, The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations: <http://www.santafe.edu/projects/swarm/overview/overview.html>. Santa Fe Institute.

[3] Terna, P. (1998), Simulation Tools for Social Scientists: Building Agent Based Models with SWARM: <http://jasss.soc.surrey.ac.uk/1/2/4.html>. *Journal of Artificial Societies and Social Simulation* vol. 1, no. 2.

[4] JAFMAS Java-Based Framework for Multi-Agent Systems.  
<http://www.ececs.uc.edu/~abaker/JAFMAS/index.html>

[5] Zannier, C., Agent Based Simulation. A report on: Modeling Multiagent Systems as Self-Organized Critical Systems. (2001) University of Calgary.