MASON: A Java Multi-Agent Simulation Library

Sean Luke, Gabriel Catalin Balan, and Liviu Panait
George Mason University
http://www.cs.gmu.edu/∼eclab

We present MASON, a new multiagent simulation library written for Java. MASON is a general-purpose, single-process, discrete-event simulation library intended to support diverse multiagent experiments ranging from 3D continuous robotics to social complexity networks to discretized ant foraging algorithms.

MASON is of special interest to the social insect algorithm community because its primary design goal is to support very large numbers of agents efficiently. As such, MASON is faster than scripted systems such as StarLogo[1] or Breve[2], while still remaining portable and producing guaranteed replicable results. In other papers submitted to this workshop, we have successfully used the system to develop by hand, and to apply evolutionary computation to search for, ant foraging behaviors involving thousands of ants and multiple pheremones.

Many multi-agent simulation environments are designed to meet the needs of a particular discipline; for example, TeamBots[3] emphasizes robotics, while RePast[4], Ascape[5], and Swarm[6] emphasize discrete environments with networks of interacting social agents. In contrast, MASON's second design goal is to make it easy to build a wide variety of multi-agent simulation environments (in our case, to test machine learning and artificial intelligence algorithms). Rather than provide an all-encompassing, rigid framework to meet this generalist criterion, MASON is a small, portable core around which specialized tools may be built for different tasks.

MASON consists of two parts: the simulator model library proper, and tools for visualizing and manipulating the model via a graphical interface. The model and the visualization libraries are completely separated. This separation fulfills a third design goal of the simulator: to run efficiently while headless on back-end server machines, but permit the experimenter to view or modify checkpointed simulations during an experimental run. The model may be serialized to and recovered from storage at any time, and the visualization system may be added or removed from the model at any point. Runs may be repeated on any platform with identical results.

MASON's model library contains a discrete-event schedule to represent time, plus various spatial representations called *neighborhoods*. MASON has no prescribed set

[1] http://education.mit.edu/starlogo
[2] http://www.spiderland.org
[3] http://www.teambots.org
[4] http://www.repast.org
[5] http://www.brook.edu/dybdocroot/es/dynamics/models/ascape/
[6] http://www.swarm.org

of spatial models: at present the library comes with plain and toroidal models for 2D discrete, 2D hexagonal, 2D continuous, 3D discrete, and 3D continuous spaces. We plan to add graph and multigraph neighborhoods. Any object may be stored in these neighborhoods, and the models may be used in any combination and any number in a given simulation. MASON separates the notion of an "agent" from embodiedness: agents are simply objects which may be scheduled to be executed. When executed, agents typically manipulate objects stored in the neighborhoods. Like any other object, agents may be embodied in the neighborhoods if this is appropriate to the simulation proper.

To enable complete separation of model from visualization, MASON adopts the notion of *portrayal* objects which are tasked to display various neighborhoods or individual objects within those neighborhoods. Portrayals also permit a user to graphically manipulate the objects and neighborhoods. The library provides basic, easily extended portrayals for all of its model environments, including ones which draw 2D models in 3D.

MASON comes with several built-in example applications, including ant foraging, flocking behaviors in continuous 2D and 3D, continuous models simulating virus infection and cooperative target observation, and 2D discrete and hexagonal heat bugs. We expect to release MASON as open source by the end of Summer, 2003.