Replication of Sugarscape Using MASON¹

Anthony Bigbee, Claudio Cioffi-Revilla, Sean Luke

Center for Social Complexity and Evolutionary Computation Laboratory
George Mason University, Fairfax, VA 22030 USA ccioffi@gmu.edu

1 Introduction

The purpose of this research was to replicate the Sugarscape model (Eptstein and Axtell 1996) and simulation outcomes as described in Growing Artificial Societies (GAS). Sugarscape is a classic agent-based model and contemporary simulation toolkits usually only have a very simple replication of a few core rules. There is scant evidence of significant replication of the rules and simulation outcomes; code supplied with Repast, Swarm, and NetLogo implement a minority of the rules in Sugarscape. In particular, the standard Repast distribution only implements Growback, Movement, and Replacement. Sugarscape implementations in these toolkits are clearly provided only as basic demonstrations of how well-known social models might be implemented, rather than complete achievements of scientific replication.

A major goal included assessing the maturity of the new MASON toolkit to replicate Sugarscape. MASON (Multiagent Simulator of Neighborhoods) “is a fast discrete-event multiagent simulation library core in Java, designed to be the foundation for large custom-purpose Java simulations, and also to provide more than enough functionality for many light-

weight simulation needs.” (Luke et al. 2005). Since MASON was designed to be a tool for social science research, among other uses, replication of one of the most recognized agent-based social science models would demonstrate its maturity and usability for its intended purpose.

Replication of well-known models is also important given the relative novelty of agent-based modeling in social science. Better tools and technique for lowering barriers to entry by social scientists are desirable outcomes.

2 Approach

Epstein and Axtell (1996) offer a framework – Sugarscape – for agent-based modeling and simulation that revolves around the following elements: agents, environment, rules. Epstein and Axtell state that the defining feature of the Sugarscape/artificial society model is that “fundamental social structures and group behaviors emerge from the interaction of individual agents operating on artificial environments under rules that place only bounded demands on each agent’s information and computational capacity.” Computationally, Sugarscape rests on an ‘object-oriented’ approach consisting of:

- **Instance variables** representing agents’ internal states or attributes (such as sex, age, wealth);
- **Methods** for agents’ rules of behavior (such as eating, trading, combat);
- **Encapsulation** of agents internal states and rules to facilitate agent-based model construction.

Details regarding object-oriented (OO) techniques in Sugarscape are generally omitted from GAS. Appendix A contains a short section on OO techniques used and considered. Polymorphism is not discussed and inheritance was considered but was not used due to “efficiency considerations …. In total, each agent has over 100 methods.” By comparison, the single agent class in MASON Sugarscape has approximately 32 methods, although only 75-80% of all Sugarscape rules were implemented. A prototype implementation of Sugarscape using ASCAPE appears to employ polymorphism and inheritance.