How Much Is Too Much? --
What Programming Skills Are Really Needed to do ABM?

Laszlo Gulyas
Computer and Automation Institute
Hungarian Academy of Sciences

Sandor Bartha
Lorand Eotvos University
Budapest, Hungary

Extended Abstract

2/28/2003

Recent discussions in the agent-based modeling community have showed disparate opinions about the level of programming skills to be expected from modelers. Everybody seems to agree that the more skill candidates have, the better. However, most of today's students lack these capabilities and developing them requires substantial efforts from the adventurous entrepreneur. Therefore, lowering the requirements would help agent-based modeling becoming a more widely accepted methodology.

Some suggest that with the spread of general computer literacy, the problem outlined above will eventually go away, sooner than we would imagine. Others argue that the problem is inherently social: mastering mathematics or statistics is not in the least easier than learning to program, still no aspiring scientist can afford to avoid it.

It would be hard to deny any of these arguments. Nonetheless, we argue that today's requirements can be lowered. This statement is not very surprising either. Various model building tools (such as NetLogo or AgentSheet) demonstrate that by limiting the 'space' of possible models, the task of modeling can be efficiently assisted. The real challenge is to bridge the gap between the potential open-ended nature of Swarm-alike modeling environments (e.g., Swarm, RePast, Ascape) and the ease of use provided by the former frameworks.

Graphical model building interfaces for general ABM platforms, such as SimBuilder (for RePast) and the Visual Swarm Builder (VSB) are attempts to achieve exactly this. Still, they impose certain limitations on the modeler and require a certain level of programming.

In this talk we speculate about possible ways to improve upon the current situation. On one hand, we argue that a large part of the observational machinery, especially the generation of statistical output, can be assisted in an interactive way. Both SimBuilder

and VSB, or Ascape and NetLogo provide examples to this effect. On the other hand, we argue that a relatively large portion of the code of today's models can be considered as "accidental representation". These are algorithms and data structures to translate the conceptual model's notions into programming language. The application of these is often not conscious, but is based upon words-of-mouth information. These are the parts of the model where programming skills count the most.

Our vision is a abstract formalism to describe agent-based models. Models defined in this language would then be automatically transformed into agent-based simulations in RePast, Swarm or Ascape, etc. Such a description would be ideal to publish concise definitions of agent-based models, and thus it could help replication efforts. Moreover, independently developed compilers to different modeling platforms in place, the formalism could make the docking of computational models a routine task.

Obviously, this is a rather ambitious vision with a number of possible caveats. For example, even if the goal is achievable, it would probably never generate code as efficient as an experienced programmer could. Still, with the use of techniques from the field of Generative Programming, the option may be viable for smaller scale, exploratory models. Also, it would force making optimizations in the code explicit with respect to the conceptual model.

Our aim with this talk is to generate discussion and to present preliminary results of our work towards such a formalism. These include experiments with mathematics-based formalisms to describe simple models, for example, the canonical Heatbugs simulation.