2

Spatial Simulation Using the SME

THOMAS MAXWELL, ALEXEY VOINOv, AND ROBERT COSTANZA

2.1 Introduction

This chapter describes an integrated environment for high-performance spatial modeling, called the Spatial Modeling Environment (SME) (Maxwell and Costanza, 1994, 1995, 1997b). This environment, which transparently links icon-based modeling environments with advanced computing resources, allows scientists to develop models in a user-friendly, graphical environment, requiring very little knowledge of computers or computer programming. Automatic code generators construct spatial simulations and enable distributed processing over a network of parallel and serial computers, allowing transparent access to state-of-the-art computing facilities.

The modeling environment imposes the constraints of modularity and hierarchy in program design and supports archiving of reusable modules in our xml-based Simulation Module Markup Language (SMML) (Maxwell, 1999; Maxwell et al., 1997a). An associated library of “module wrappers” will facilitate the incorporation of legacy simulation models into the environment. This paradigm encourages the development of libraries of modules representing model components that are globally available to model builders, enabling users to build on the work of others instead of starting from scratch each time a new model is initiated.

The SME design has arisen from the need to support collaborative model building among a large, distributed network of scientists involved in creating a global-scale ecological/economic model. The effort to provide integrated support for graphical icon-based unit model development, modular model development, and high-performance computing and visualization has led to the formulation of a three-part Constructor–Modelbase–Driver architecture (Fig. 2.1). The SME architecture consists of four primary applications: the Module Importer, the Code Generator, the SME Driver, and the Java portal. The user may invoke these applications either through a command line interface or using the SME graphical user interface. These components are introduced here and described in more detail in the following sections.
**Constructor:** The Constructor component of the SME is used to graphically construct, calibrate, and test dynamic process-based modules. These modules define the processes (biological, ecological, socioeconomic, etc.) occurring within a single cell of a spatial model. The Constructor component is represented by an off-the-shelf graphical modeling tool such as Stella, Extend, or PowerSim.

**Modelbase:** The Module Importer translates the Constructor ecosystem component modules into generic module specifications defined in our xml-based SMML. The SMML modules can be archived in the Modelbase to be accessed by other researchers and/or linked to specify a unit model describing the dynamics of a single cell within the spatial model.

**Code Generators:** The Code Generator uses the unit model specification in SMML to generate the C++ code describing a spatial model. At this stage, the unit-cell models are replicated to create a spatial grid covering the study area. This simulation code is incorporated into the SME Driver, a distributed simulation application. The user customizes the generated model by providing configuration information.

**Driver:** The SME Driver is a distributed object-oriented simulation environment that incorporates the set of code modules that actually perform the spatial simulation on the targeted platform. It enables distributed processing over a network of parallel and serial computers, allowing transparent access to state-of-the-art computing facilities. It is implemented as a set of distributed C++ objects linked by message passing.

The SME environment includes a Java interface, or “portal,” that provides the user with a single familiar environment in which to build, configure, interactively run, and visualize models on any one of a number of parallel or serial computers. It facilitates remote Web-based access to spatial models running on a simulation server.