Environmental Modelling & Software 26 (2011) 1052–1055

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Short communication

A generic framework for land-use modelling

Christian Schweitzer*, Joerg A. Priess, Subhashree Das

Helmholtz Centre for Environmental Research – UFZ, Department Computational Landscape Ecology (CLE), Permoserstrasse 15, D-04318 Leipzig, Germany

ARTICLE INFO

Article history:
Received 25 February 2010
Received in revised form
16 February 2011
Accepted 26 February 2011
Available online 6 April 2011

Keywords:
Integrated framework
Map comparison
Genetic algorithm
Cellular automata
Land-use model

ABSTRACT

In this paper we present the generic modelling system SITE (SImulation of Terrestrial Environments), a software package to develop and apply models simulating regional land-use dynamics. The modelling system includes (i) a framework managing the model generics and (ii) code templates for the development of rule-based land-use and land-cover change (LUCC) models. SITE comprises built-in methods for e.g. map comparison, model optimization and environmental scenarios.

Software availability

Name of software: SITE – SImulation of Terrestrial Environments
Developers: Mimler M., Priess J.A., Schweitzer C., Das S.
Contact address: Helmholtz Centre for Environmental Research – UFZ, Department Computational Landscape Ecology, Permoserstrasse 15, D-04318 Leipzig, Germany. christian.schweitzer@ufz.de
Year first available: 2007
Hardware required: PC
Software required: Python 2.3, MySQL, Win32
Program language: C++, Python
Program size: 6 MB (Windows installer – including example Python scripts, example data)
Availability: free (upon request)
Software homepage: http://www.ufz.de/index.php?en=19080

1. Introduction

Modelling land-use, land-cover and environmental change is a field of increasing importance and a broad range of models has been developed for this discipline (Haase and Schwarz, 2009; Schaldach and Priess, 2008). For numerous modelling approaches the development or adaptation of the appropriate software is a crucial step for successfully answering research questions, a step strongly constrained by financial resources and time available for model development. In order to overcome this problem, researchers often tend to modify existing or create new software, which may be functional, but often not very intuitive in terms of applicability and transferability. For users with little or no programming skills the application of such models is difficult and a reason for little or no interest in those approaches. To address this problem, modelling frameworks try to bridge this gap by providing convenient user interfaces that communicate with the complex software generics. A 'framework', as it is used in our context, is defined as generic software that can be reused for custom applications, with the advantage that the software can be developed several times faster than without the framework (Fayad et al., 1999). In this manuscript we present the key features of the SITE framework and show an implementation example for a rule-based land-use model, in which land allocation is based on a multi-criteria analysis and regional preferences and constraints.

2. Modelling system

From the beginning, a core objective of the SITE developers was to facilitate maintenance and reuse of components within the entire software package, as the tool was intended to be applicable in more than one case study (Mimler and Priess, 2008). SITE consists of two main parts (i) the system domain (SD), which includes optimized methods, procedures and essential tools for the modelling process, implemented in C++ and (ii) the application domain (AD), which is a Python interface designed for the

* Corresponding author. Tel.: +49 341 235 1962; fax: +49 341 235 3939.
E-mail address: christian.schweitzer@ufz.de (C. Schweitzer).

1364-8152/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.envsoft.2011.02.016

implementation of modelling routines and decision rules to address land-use modelling issues (Fig. 1).

SITE currently uses the concept of cellular automata (CA) models, which are defined as spatially and temporary discrete systems consisting of cells arranged in a lattice of n (>1) dimensions. The cells represent a discrete moment in time and change their current state due to a set of rules, mostly deterministically formulated in a local transition function taking into account the current state of neighbouring cells (Schiff, 2010). Theoretically it is possible to represent land-use decisions in SITE with spatial non-CA models such as multi-agent systems (Berger et al., 2006) or CLUE-type regression equations (Veldkamp and Fresco, 1996), although in the latter case much of the flexibility of rule-based approaches would be lost.

SITE has been applied for three case studies addressing different land-use related research questions. The first application investigated the land-use dynamics of tropical rainforest margins in Sulawesi, Indonesia (Priess et al., 2007). The second and third case studies, which are under advanced stages of development, analyse the impacts of LUCC on water resources and impacts of wildfire in Mongolia (Priess et al., 2010; Schweitzer and Priess, 2010) and the potential of bioenergy crops and food-fuel conflicts in South-India (Das et al., 2010), respectively. Fig. 2 presents a generalized view of the SITE components used in previous studies. A detailed description of the software implementation is given in Mimler (2007) and Mimler and Priess (2008).

2.1. System domain

The SD comprises the generic aspects in SITE and handles the grid system, cell-attributes, cell-/cluster-iterators and other components. The latter integrates methods for model testing and calibration using map comparison algorithms, distance and neighbourhood operations and clustering procedures. SITE includes a graphical user interface (GUI), which supports the execution of land-use models and visualisation of simulation results. A 3-D view offers the possibility to freely assign (and combine) x/y- and z-axis with cell-attributes (e.g. ‘crop x’ and ‘distance to market’). Additionally, a command line is available for consecutive runs, e.g. during sensitivity analysis or model calibration.

The coupling of third-party models is enhanced by providing a generic interface that allows the implementation of feedbacks between different framework components and different processes, which are considered as an essential aspect of integrated approaches (Verburg, 2006). In recent studies we linked the biophysical model DayCent (Parton et al., 1998) to simulate carbon dynamics, biomass and crop yields (Priess et al., 2007, 2010; Schweitzer and Priess, 2010; Das et al., 2010). The integration is based on a client-server solution in which SITE acts as client

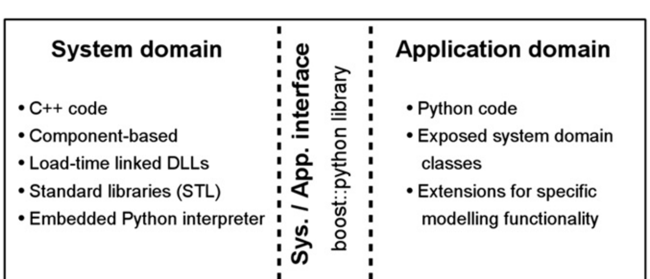

Fig. 1. Software details of the system and application domain. The system domain includes generic software components that handle and execute models, while case study specific code is located in the application domain (Mimler, 2007).

distributing modelling jobs (= annual simulations for cells or clusters) to the DayCent-server. The interface can configure and execute jobs and handle incoming simulation results.

2.2. Application domain

The AD is designed as a scripting interface to access functionalities and methods implemented in the SD. Setting up a new model in the AD requires minimally the use of the two functions *Initialize*() and *SimulationStep*(), in which the model's scheduling has to be included. The sequence of sub-module calls reflects the model's scheduling hierarchy. Potential users benefit from the existing case studies, which already provide a set of sub-modules that can be modified to cover new research objectives. Fig. 3 presents a generalized scheduling structure within a Python file that can be executed by SITE, based on the ones we use in current studies.

2.2.1. Initialization

Initialize() first generates the simulation grid, followed by operations setting the cells to an initial status with cell-specific attribute values (e.g. count the number of neighbour cells, perform distance calculations etc.). To reflect identical or almost identical biophysical properties of grid cells and simultaneously reduce the high runtime requirements of DayCent, the cluster algorithm of SITE groups cells of the same land-use or land-cover type to derive units of uniform geographical and biophysical properties, comparable to hydrological response units – HRUs, a concept often applied in hydrological studies (Flügel, 1995). Cluster definitions (and resulting cluster size) usually depend on data quality, research question and may apply any cell attribute present in the case study.

2.2.2. Simulation step

In *SimulationStep*(), land-use decisions are simulated typically once a year following a three step process: (i) suitability analysis based on a multi-criteria approach, (ii) execution of allocation modules, that are driven by the demand for commodities (e.g. space for housing, food, wood) and (iii) calculation of plant growth, biomass and trace gas fluxes both for natural vegetation and agro-ecosystems, currently using DayCent.

2.2.2.1. Suitability analysis. The task of the suitability module (SUIT) is the generation of dynamic suitability maps for each of the land-use and land-cover classes expected to change. SUIT employs a multi-criteria approach which is transparent, flexible and has the capacity to integrate large amounts of heterogeneous data (Eastman et al., 1995). SUIT is subdivided into functions computing biophysical suitabilities (e.g. elevation, terrain slope, soil fertility, precipitation) and suitabilities based on socio-economic factors, if necessary (e.g. gross margins, accessibility, farmers' preferences). All suitability values are normalised to a range between 0 (not suitable) and 1 (perfectly suitable) following the equation:

$$s_{kl} = \underbrace{\left(w_{\mathrm{B}} \sum_{i=1}^{m} \beta_i s_{Bikl} + w_{\mathrm{E}} \sum_{i=1}^{n} \epsilon_i s_{Eikl}\right)}_{\text{suitability}} \cdot \underbrace{\prod_{j=1}^{o} c_{Bjkl} \prod_{j=1}^{p} c_{Ejkl}}_{\text{constraints}} \qquad (1)$$

with $w_{\mathrm{B}} + w_{\mathrm{E}} = 1$; $\sum_{i=1}^{m} \beta_i = 1$; $\sum_{i=1}^{n} \epsilon_i = 1$; $s_{Bikl}, s_{Eikl}, c_{Bjkl}, c_{Ejkl} \in [0,\ 1]$

The calculation of the overall suitability value s_{kl} for each grid cell k and land-use type l consists of two terms. In first part (suitability) the mean value of the partial suitabilities s_{Bikl} for biophysical and s_{Eikl} for socio-economic criteria are weighted using the partial weights β_i/ϵ_i, where m/n represent the total number of