ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

SoftwareX ▮ (▮▮▮▮) ▮▮▮–▮▮▮

SoftwareX

www.elsevier.com/locate/softx

Ocelet: Simulating processes of landscape changes using interaction graphs

P. Degenne, D. Lo Seen*

Cirad, Environment and Societies Department, UMR TETIS, Montpellier, France

Received 9 February 2016; received in revised form 6 April 2016; accepted 10 May 2016

Abstract

This paper introduces Ocelet, a domain specific language and simulation tool for modelling changes in geographical landscapes. It is characterised by the use of *interaction graphs* (graphs with interaction functions on their edges) to represent the system as composed of processes, each involving several entities distributed in space that are in interaction with each other. Entities are the vertices of the graphs, and interactions are the edges on which (interaction) functions can be applied to make the system change through time. Examples are given to illustrate the generic disposition of the simulation approach to model and study changing geographical setups.

Keywords: Spatial modelling; Multi-scale; Domain specific language

Code metadata

Current code version	*v1.1*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00024
Legal Code License	*CeCILL v2.1(GNU GPL compatible Free Software licence agreement designed byFrench Public Research Institutes)* Licence detailed information: http://www.cecill.info/licences/Licence_CeCILL_V2.1-en.html
Code versioning system used	*Git*
Software code languages, tools, and services used	*Java 7, Eclipse, Xtext*
Compilation requirements, operating environments & dependencies	*Development and compilation : Eclipse Luna, Java 7* *Dependencies : Eclipse RCP, Xtext-XBase 2.8, Geotools 9.4, JAK*
If available Link to developer documentation/manual	*NA*
Support email for questions	*ocltdev@ocelet.fr*

Software metadata

Current software version	*Carmine (1.1)*
Permanent link to executables of this version	*Download tab on* http://www.ocelet.org/
Legal Software License	*CeCILL V2.1*
Computing platforms/Operating Systems	*Linux, OS X, Windows*
Installation requirements & dependencies	*Java 7 (or 8) Runtime Environment*
If available, link to user manual - if formally published include a reference to the publication in the reference list	Documentation tab on http://www.ocelet.org/
Support email for questions	*ocltdev@ocelet.fr*

1. Motivation and significance

Simulation models are used both as research tools, to test hypotheses when trying to improve the understanding of a

* Corresponding author.
E-mail addresses: loseen@teledetection.fr, danny.lo_seen@cirad.fr (D. Lo Seen).

http://dx.doi.org/10.1016/j.softx.2016.05.002

Please cite this article in press as: Degenne P, Lo Seen D. Ocelet: Simulating processes of landscape changes using interaction graphs. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.05.002

system, and in decision support, to explore alternative scenarios (e.g. [1]). But modelling the environment as a system can be considered particularly challenging as several interacting processes often need to be modelled. These processes can also be dynamic, spatially distributed, at several scales of space and time, and may involve human activities [2]. Several reviews of available methods for modelling the environment exist (e.g. [3–5]). Most of the methods belong to three main approaches that stand out by the size of their user communities: Systems Dynamics (SD), Cellular Automata (CA) and Agent-based modelling (ABM). The SD approach proposed by Forrester [6,7] represents real-world processes in terms of stocks (system variables), flows (exchanges between stocks) and interacting feedback loops (an output of the system can be fed back as input to the system). Examples of software based on these principles include STELLA [8] and Vensim [9]. But when a system is distributed in a geographical space, aggregated system variables become inadequate. The solution proposed by the Spatial Modelling Environment (SME—[26]) was to disaggregate the system space into cells. Stock-flow models could then be included in each of the cells, with neighbouring cells able to exchange flows. Referred to as "individual-based modelling approaches", CA and ABM are inherently different in that aggregate patterns emerge from the sum of individual behaviour [10]. With CA, geographical space is represented by grid cells that can take a finite number of states. The state of a given cell changes following transition rules that depend on the states of the neighbouring cells. Urban dynamics is a field where CA application has been particularly successful (e.g. DEUM—[11]; SLEUTH—[12], and more recently O'Sullivan, 2001; [13]). When the system to be modelled involves heterogeneous entities in more complex situations such as those in social systems, ABM is generally preferred. Agents are defined by their behaviour, can be reactive or cognitive, and interact with other agents and their environment [14]. A review of the use of ABM in ecosystem management can be found in [15]. Software for multi-agent simulations includes CORMAS (Bousquet et al., 1998), NetLogo [16] and GAMA [17].

All three modelling approaches have specific characteristics that can be considered merits or weaknesses depending on the objectives sought. In particular, modellers often need to study a system as a whole, and at the same time decipher how local and intermediate level processes sum up to form the whole system. It is therefore not surprising that there have been attempts to mix or integrate the different approaches. For example, the SME mentioned above can be considered an integration of SD and CA, whereas Clarke [10] explored the origins and key respective contributions of CA and ABM. Schieritz and Milling [18] carried out a detailed comparison of SD and ABM, and reflected on previous promising but still unfulfilled attempts to combine top-down (SD) and bottom-up (ABM) approaches. Since 2008, we have been developing an approach that can be considered intermediate between top-down and bottom-up approaches. The rationale was not to integrate the two types of approaches but rather to focus on their "common denominator" that are the interactions. Any system is described in terms of entities distributed in space that are in interaction with each other, and simulation models of geographical changes are built using **interaction graphs** to explicitly describe processes [19]. The interaction graphs have entities as vertices, and interaction functions attached to their edges. A graph alone can only structure the neighbourhood relationships in a system (which entity is in relation with which other entity) and not the nature of the relations (what happens when entities interact). Nor does it describe how the system evolves with time. Interaction graphs were thus introduced as an extension to the mathematical definition of graphs by allowing (interaction) functions to be applied on the edges simultaneously [20]. These functions are able to access the properties of the entities connected, use and optionally change them, according to the processes being modelled. The interaction graphs are also dynamic in the sense that vertices and edges can be added or removed, and their properties modified during the simulation.

When modelling a system and its dynamics using interaction graphs, one has to imagine what interactions are at play in the system, how they are distributed (spatially, functionally, socially...) and how they can influence the temporal evolution of that system. Such a definition of an interaction graph is very generic. One same concept is used to describe hierarchical relationships (allowing aggregation and disaggregation operations), spatial relationships (from regular grid based neighbourhood, to any other structure issued from vector based geographic information layers or from a continuous spatial reference system), social relationships (by writing socially meaningful semantics in the interaction functions), or more generally any kind of functional based relationship. We combine this genericity with well-chosen operators in the form of a Domain Specific language (see [21], for a review of DSL in ecological modelling) to offer a rich capacity of expression for modelling a wide range of spatially explicit systems and their dynamics. The generally spatial entities used in Ocelet models are represented with data types that are commonly used in Geographical Information Systems (GIS): points, lines, polygons, multi-points, multi-lines and multi-polygons. Interactions between entities result in changes in the state and (spatial) configuration of the entities. The key difference is that, once imported from a GIS data file (e.g. shapefile) into the model, the entities no longer belong to a "GIS layer", and can be interconnected individually through several interaction graphs.

Spatial dynamics models are built within a software environment called the "Ocelet Modelling Platform" (OMP). After a few years of practice and the transition phase between the initial prototype and the current stabilised version of Ocelet (version 1.1), we hereby (Section 2) present the main concepts and features of the software. Three test cases are then briefly described to illustrate the generic disposition of Ocelet (Section 3). Finally, we discuss how the approach can contribute to address scientific questions and also what are the main types of modelling situations that can be tackled (Section 4).

2. Software description

The OMP software environment is built around the Ocelet DSL in order to facilitate model creation and maintenance, code