



Software, data and modelling news

StellaR: A software to translate Stella models into R open-source environment

Babak Naimi^{a,*}, Alexey Voinov^b^a Department of Environment and Energy, Science and Research Branch, Islamic Azad University, Tehran, Iran^b Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Article history:

Received 22 March 2012

Accepted 18 May 2012

Available online 15 June 2012

Keywords:

Stella

Modelling

Dynamic simulation

R-project

Open source

ABSTRACT

Stella is a popular system dynamics modeling tool, which helps to put together conceptual diagrams and converts them into numeric computer models. Although it can be very useful, especially in participatory modeling, it lacks the power and flexibility of a programming language. This paper presents the StellaR software which translates a Stella model into a model in R, an open source high level programming language. This allows using conceptual modeling tools provided in Stella, together with computational functionality and programming flexibility provided in R. It also opens access to powerful software libraries available in R, which is especially useful for spatial modeling.

© 2012 Elsevier Ltd. All rights reserved.

Software availability

Name of software: StellaR

Developer: Babak Naimi, Alexey Voinov

Download: <http://www.r-gis.net/stellar>

Year first available: 2011

Program language: Python

Program size: 3.0 MB

Availability and cost: Open source

The system dynamics simulation approach helps to understand the complex interrelationships existing between different components within a system. Several well-established dynamic simulation systems including Stella, PowerSim, Berkeley Madonna, Vensim, Simile, etc. offer similar features and have been used for modeling studies. Stella was one of the first dynamic modeling systems to achieve broad recognition and use, due to its user-friendly graphic interface. One of the reasons that system dynamics software became so popular in modeling is that they provide handy tools to put together conceptual diagrams, and provide tools to convert them into numeric computer models. Although they can be very useful for specific applications they often lack the full power and flexibility of a programming language (Petzoldt and Rinke, 2007). They are also quite restrictive for spatial dynamics and have limited functionality for model analysis (calibration, statistical analysis, etc.). Besides, they are proprietary and

can hardly be modified to meet the needs of a particular modeling exercise. While the Graphic User Interface (GUI) is a big plus, the fact that the systems are closed and the model code is either not available or difficult to use outside of the interface becomes restricting in case of more complex, especially spatially explicit models. Some attempts have been made previously to link the system dynamics GUI with real programming. For example the Spatial Modeling Environment (SME) (Maxwell et al., 2003) carried much promise. However with discontinued support, it quickly becomes outdated and hardly usable.

At the same time new modeling environments appear, such as R (R Development Core Team, 2011), which is a general-purpose high level programming language and a free (under the GNU general public license) and open source environment. It can be used for scientific computing, data analysis and graphical visualization. Although R has been widely known as an environment for statistical analysis and graphical representation, it is increasingly becoming more suitable for general mathematical computing (Soetaert et al., 2010a). The capabilities of R can be extended through user-created packages, which allow developing new and specialized analytical techniques, graphical devices, import/export capabilities, reporting tools, etc. All of these capabilities make R very powerful and the number of its users and its functionality is growing exponentially. An increasing collection of packages explicitly developed for dynamic modelling (e.g. deSolve, simecol, FME, ReacTran, etc.) (Soetaert et al., 2010a) makes the R system a suitable environment for the development, implementation and analysis of dynamic models.

* Corresponding author.

E-mail addresses: naimi@itc.nl, naimi.b@gmail.com (B. Naimi).

StellaR provides a bridge to use conceptual modeling tools provided in Stella, together with superb computational functionality and programming flexibility provided in R. StellaR can translate a Stella model into a model in the R programming language. The R model uses the R package *deSolve* (Soetaert et al., 2010b) to solve initial value problems formulated as ordinary differential equations (ODE), differential algebraic equations (DAE) and partial differential equations (PDE). StellaR can run on popular platforms (e.g. MS Windows, Linux, Mac). StellaR uses a command-line interface and needs two arguments to work. First argument is the name of a text file containing the equations exported from Stella, and the second argument is the name for the model in R.

StellaR uses a text processing procedure to identify different components in the Stella model including differential equations, flows, convertors, data, functions and parameters. Object oriented approach was used to define classes for model, flow and convertor. Two approaches were used to translate the built-in functions in Stella. For those functions that have equivalents in R, the syntax of the function is translated to match the syntax of the function in R. For those functions that do not exist in R (e.g. *SINWAVE*), its equivalent R function is developed and included in a separate R script file, which can be called and used by the R model. Data are

translated and converted into a file with comma separated values (csv). The data file is read by the model in the R environment. The parameter names are kept the same as in the original Stella model. The model including the differential equations and the defined processes (flows) is specified as a function in R. Input to the function (model) is the model time, the initial values of the state variables and the parameters. This function will be called by the ODE function in *deSolve* package (Soetaert et al., 2010a, 2010b). See StellaR web site for more details, examples and instructions.

References

- Maxwell, T., Voinov, A.A., Costanza, R., 2003. Spatial simulation using the SME. In: Costanza, R., Voinov, A.A. (Eds.), *Landscape Simulation Modeling: A Spatially Explicit, Dynamic Approach*. Springer, Berlin, pp. 21–42.
- Petzoldt, T., Rinke, K., 2007. Simecol: an object-oriented framework for ecological modeling in R. *Journal of Statistical Software* 22 (9), 1–31.
- R Development Core Team, 2011. *R: a Language and Environment for Statistical Computing*. Vienna, Austria.
- Soetaert, K., Meysman, F., Petzoldt, T., 2010a. Solving differential equations in R. In: Psihoyios, G., Tsitouras, C. (Eds.), *Numerical Analysis and Applied Mathematics*, vols. I–III. Amer Inst Physics, Melville, pp. 31–34.
- Soetaert, K., Petzoldt, T., Setzer, R.W., 2010b. Solving differential equations in R: package *deSolve*. *Journal of Statistical Software* 33 (9), 1–25.