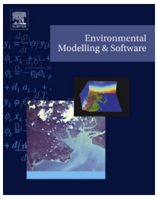
Environmental Modelling & Software xxx (2014) 1–8

Contents lists available at ScienceDirect

Environmental Modelling & Software

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



Serving many at once: How a database approach can create unity in dynamical ecosystem modelling☆

Wolf M. Mooij [a,b,*], Robert J. Brederveld [c], Jeroen J.M. de Klein [b], Don L. DeAngelis [d], Andrea S. Downing [e], Michiel Faber [c], Daan J. Gerla [f,p], Matthew R. Hipsey [g], Jochem 't Hoen [b], Jan H. Janse [h], Annette B.G. Janssen [a,b], Michel Jeuken [i], Bob W. Kooi [j], Betty Lischke [k], Thomas Petzoldt [l], Leo Postma [h], Sebastiaan A. Schep [c], Huub Scholten [m], Sven Teurlincx [a], Christophe Thiange [h], Dennis Trolle [n], Anne A. van Dam [o], Luuk P.A. van Gerven [a,b], Egbert H. van Nes [b], Jan J. Kuiper [a,b]

[a] *Department of Aquatic Ecology, Netherlands Institute of Ecology (NIOO-KNAW), P.O. Box 50, 6700 AB Wageningen, The Netherlands*
[b] *Aquatic Ecology and Water Quality Management Group, Department of Environmental Sciences, Wageningen University, P.O. Box 47, 6700 AA Wageningen, The Netherlands*
[c] *Witteveen+Bos, P.O. Box 233, 7400 AV Deventer, The Netherlands*
[d] *USGS/Biological Resources Division and Department of Biology, University of Miami, P.O. Box 249118, Coral Gables, FL 33124, USA*
[e] *Department Systems Ecology, Stockholm University, 10691 Stockholm, Sweden*
[f] *Department of Ecosystems, Institute for Marine Resource and Ecosystem Studies (IMARES), PO Box 167, 1790 AD Den Burg, The Netherlands*
[g] *University of Western Australia, School of Earth and Environment, Crawley, WA 6009, Australia*
[h] *PBL, Netherlands Environmental Assessment Agency, P.O. Box 303, 3720 AH Bilthoven, The Netherlands*
[i] *Deltares, P.O. Box 177, 2600 MH Delft, The Netherlands*
[j] *Department of Theoretical Biology, Faculty of Earth and Life Sciences, VU University, De Boelelaan 1085, 1081 HV Amsterdam, The Netherlands*
[k] *Department of Ecology and Ecosystem Modelling, Institute of Biochemistry and Biology, University of Potsdam, Am Neuen Palais 10, 14469 Potsdam, Germany*
[l] *Faculty of Environmental Sciences Institute of Hydrobiology, Technische Universität Dresden, 01062 Dresden, Germany*
[m] *Information Technology Group, Department of Social Sciences, Wageningen University, P.O. Box 8130, 6700 EW Wageningen, The Netherlands*
[n] *Department of Bioscience, Aarhus University, Vejlsøvej 25, 8600 Silkeborg, Denmark*
[o] *UNESCO-IHE Institute of Water Education, 2601 DA Delft, The Netherlands*
[p] *Department of Ecosystem Studies, Royal Netherlands Institute for Sea Research (NIOZ), PO Box 140, 4400 AC Yerseke, The Netherlands*

ARTICLE INFO

Article history:
Received 30 June 2013
Received in revised form
1 April 2014
Accepted 3 April 2014
Available online xxx

Keywords:
Modelling framework
Programming language
Differential equation
Community-based modelling
Database approach to modelling
DATM

ABSTRACT

Simulation modelling in ecology is a field that is becoming increasingly compartmentalized. Here we propose a Database Approach To Modelling (DATM) to create unity in dynamical ecosystem modelling with differential equations. In this approach the storage of ecological knowledge is independent of the language and platform in which the model will be run. To create an instance of the model, the information in the database is translated and augmented with the language and platform specifics. This process is automated so that a new instance can be created each time the database is updated. We describe the approach using the simple Lotka–Volterra model and the complex ecosystem model for shallow lakes PCLake, which we automatically implement in the frameworks OSIRIS, GRIND for MATLAB, ACSL, R, DUFLOW and DELWAQ. A clear advantage of working in a database is the overview it provides. The simplicity of the approach only adds to its elegance.

1. Introduction

Since the onset of ecological simulation modelling based on differential equations – in the sixties and seventies of the last century – attempts have been made to bring conceptual unity through the development of modelling frameworks. In the field of

☆ Thematic Issue on the Aquatic Ecosystem Models.

* Corresponding author. Department of Aquatic Ecology, Netherlands Institute of Ecology (NIOO-KNAW), P.O. Box 50, 6700 AB Wageningen, The Netherlands. Tel.: +31 6 3177 1708.

E-mail address: w.mooij@nioo.knaw.nl (W.M. Mooij).

http://dx.doi.org/10.1016/j.envsoft.2014.04.004

aquatic ecology, such frameworks include the widely used DELWAQ – a library of water quality and ecology models developed by Delft Hydraulics (Delft Hydraulics, 1995; Deltares, 2013), as well as the Computational Aquatic Ecosystem Dynamics Model (CAEDYM) – a library of ecological process sub-models (Hipsey et al., 2007), AQUASIM (Reichert, 1994), the Dutch Waterboards' DUFLOW framework (Spaans et al., 1989) and the recently developed FABM – Framework for Aquatic Biogeochemical Models (http://fabm.sourceforge.net). Each of these frameworks is internally consistent, intuitive and well suited to answer the ecological questions it was designed for (Clemmens et al., 1993; Gal et al., 2004), and all are based on the same basic mathematical principles underlying the differential equations. Nonetheless, because these frameworks were developed independently, they all have their own sets of implementation requirements, language and coding specifications, spatial configuration options as well as boundary conditions and forcing function specifications, etc.

A user must therefore invest a considerable amount of effort to master any given framework, which in turn reduces the number of frameworks that any single user can master. The choice of framework to be used for any given project is thus primarily based on its availability, owned licenses, user experience and developer familiarity. This in turn leads to models being locked into their given frameworks, a narrowing-down of scientific expertise to the framework-scale and to the proverbial 're-invention of the wheel' – i.e., the inefficient redevelopment of existing tools for each framework, rather than a more productive cross-pollination of approaches to analyze models across frameworks, institutions, disciplines and scientists (Leavesley et al., 2002; Mooij et al., 2010; Trolle et al., 2012). We are confronted with the paradoxical situation that, while there is unity within each framework, there is no unity at the level of the ecological models.

Here we propose a method to bring unity at the level of the ecological module, with the idea that many of the existing frameworks will continue to coexist, and that, taken together, they provide the user with a wide and rich array of tools for model analysis. We coin this method a 'Database Approach To Modelling' (DATM). We developed this approach for the ecosystem model for shallow lakes PCLake, and its twin model for linear waters PCDitch. However, our approach is in no way limited to these models. In fact, it applies to all models based on differential equations and probably even beyond. We here show how one can automatically link these models to a wide variety of frameworks, including OSIRIS (Mooij and Boersma, 1996), GRIND for MATLAB (available on http://www.sparcs-center.org/grind.html), ACSL (Mitchell and Gauthier, 1976), R (R Development Core Team, 2008), DUFLOW (Spaans et al., 1989) and DELWAQ (Deltares, 2013). Note that the latter two frameworks are spatially explicit and therefore are formulated in terms of partial differential equations (PDE's), whereas implementations of an ecological model (e.g. PCLake) in the general purpose frameworks are a set of ordinary differential equations (ODE's). We will show that with DATM we can overcome this difference, and translate a single code either in a set of ODE's in a general purpose framework or as the ecological component of a set of PDE's in these spatially explicit frameworks. In the latter case, these ecological components are then merged by the frameworks with the advective and diffusive transport of matter to get the full PDE. Please note that in its current form, DATM does not provide the spatial configuration of the model, this has still to be entered at the level of the framework.

To explain the principles of DATM, we use as an example the classical Lotka–Volterra equations. These equations represent the earliest use of coupled differential equations in ecology (Lotka, 1920; Volterra, 1926, 1931). With this example, we show how knowledge of quite a few framework-specific details is necessary to implement even this simplest of models in some of the most widely used mathematical frameworks. From experience, we have learned how implementing more complex models in more specific frameworks takes a considerable effort, which is why we propose to automate this process: an essential component of DATM is the set of translators developed to automatically convert the database definitions of a given model into a working implementation in a specific framework. Conceptually, we argue that the overview and insight that arises when the model definition is stored in the database, conveniently displayed in tables and accessed through queries, facilitates model development and understanding.

2. Methods

DATM is based on the notion that ecological models are essentially rooted in mathematics. Here, we focus on models based on the mathematical concept of coupled differential equations. The dynamic systems represented by these equations have a universal mathematical notation. As an example, the Lotka–Volterra predator-prey equations can be read and understood by all in the following form:

$$\mathrm{d}V/\mathrm{d}t = r\,V - a\,V\,P \tag{1a}$$

$$\mathrm{d}P/\mathrm{d}t = a\,e\,V\,P - \mathrm{d}P \tag{1b}$$

with state variables V for prey and P for predator; parameters r for autonomous growth rate of the prey; a the attack rate of the predator on the prey, e the conversion efficiency of the predator and d the autonomous death rate of the predator. This system is in this form fully defined and ready for simulation for a given set of parameters r, a, e and d and initial conditions $V_{t=0}$ and $P_{t=0}$. Our central point is that this mathematical notation for complex simulation models is sufficient to achieve unity and transparency in ecological modelling.

As shown in the above example, the set of coupled Equations (1a) and (1b) must be augmented with information on the interpretation of the various identifiers that are used in the model. As a minimum description, the identifiers must belong to a certain class (e.g. state variable, parameter); represent a specific component of the system (e.g. prey, predator); have units (e.g. biomass, number of individuals), and (initial) values. In scientific papers that document smaller models, such as the Lotka–Volterra model, this information is often organized in tables, with either a shared table for all identifiers or separate tables per class of identifiers. Given the number of identifiers in the more complex water quality models, we choose to work with separate tables for each class of identifiers. For the Lotka–Volterra model such tables could look like (note the 's' prefix to identifiers of state variables):

Table 1
State variables.

Identifier	Description	Dimension	Initial value
sV	Prey density	Biomass *V*	(Some number)
sP	Predator density	Biomass *P*	(Some number)

for the states,

Table 2
Parameters.

Identifier	Description	Dimension	Value
r	Prey growth rate	Time^{-1}	(Some number)
a	Predator attack rate	Time^{-1} biomass P^{-1}	(Some number)
e	Predator efficiency	Biomass *P* biomass V^{-1}	(Some number)
d	Predator death rate	Time^{-1}	(Some number)

for the parameters and

Table 3
Derivatives.

Identifier	Description	Dimension	Equation
d*V*	Prey derivative	Biomass *V* time^{-1}	$\mathrm{d}V = r \cdot sV - a \cdot sV \cdot sP$
d*P*	Predator derivative	Biomass *P* time^{-1}	$\mathrm{d}P = a \cdot e \cdot sV \cdot sP - \mathrm{d} \cdot sP$

Please cite this article in press as: Mooij, W.M., et al., Serving many at once: How a database approach can create unity in dynamical ecosystem modelling, Environmental Modelling & Software (2014), http://dx.doi.org/10.1016/j.envsoft.2014.04.004