Environmental Modelling & Software 68 (2015) 110–121

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

A lightweight framework for rapid development of object-based hydrological model engines

David Kneis [1]

University of Potsdam, Institute of Earth and Environmental Sciences, Germany

ARTICLE INFO

Article history:
Received 13 October 2013
Received in revised form
9 February 2015
Accepted 12 February 2015
Available online 6 March 2015

Keywords:
Modeling framework
Generic model
Hydrology
ECHSE

ABSTRACT

Computer-based simulation models are frequently used in hydrological research and engineering but also in other fields of environmental sciences. New case studies often require existing model concepts to be adapted. Extensions may be necessary due to the peculiarities of the studied natural system or subtleties of anthropogenic control. In other cases, simplifications must be made in response to scarce data, incomplete knowledge, or restrictions set by the spatio-temporal scale of application. This paper introduces an open-source modeling framework called ECHSE designed to cope with the above-mentioned challenges. It provides a lightweight infrastructure for the rapid development of new, re-usable simulation tools and, more importantly, the safe modification of existing formulations. ECHSE-based models treat the simulated system as a collection of interacting objects. Although feedbacks are generally supported, the majority of the objects' interactions is expected to be of the feed-forward type. Therefore, the ECHSE software is particularly useful in the context of hydrological catchment modeling. Conversely, it is unsuitable, e. g., for fully hydrodynamic simulations and groundwater flow modeling. The focus of the paper is put on a comprehensible outline of the ECHSE's fundamental concepts and limitations. For the purpose of illustration, a specific, ECHSE-based solution for hydrological catchment modeling is presented which has undergone testing in a number of river basins.

Availability of the software

The ECHSE software is freely available to interested users and developers under the terms of the MIT license. The website of the project is http://echse.bitbucket.org. From this site, one can download the recent documentation files covering (1) the basic design and use of the modeling framework, (2) the installation of model engines, (3) the existing classes for rainfall–runoff modeling, as well as (4) theory and use of pre- and post-processing utilities (Kneis, 2012a,b,c,d).

The mentioned website also provides a link to the ECHSE's code repository. By cloning the repository, one gets access to

1. the latest generic source code (see Section 3.2.4),
2. the declaration and code of a set of classes for hydrological catchment modeling (see Section 4.2),
3. model engine definition files specifying the classes used by a particular engine,
4. scripts for code generation and compilation,
5. the code generator and other utilities (as self-documenting R-packages or C++ sources),
6. the documentation files.

The primary way to contribute to the project is to improve the existing classes or to design new classes and model engines being of potential interest to the community. As opposed to the application-specific classes, it is not planned to permanently maintain different forks (i. e. concurrent versions) of the software's generic core.

1. Introduction

1.1. Terminology

The term *model* is used in many different ways and contexts. To minimize the chance of confusion, the following convention is used throughout this paper:

E-mail address: david.kneis@tu-dresden.de.
[1] Present address: Technical University Dresden, Institute of Hydrobiology, Germany.

http://dx.doi.org/10.1016/j.envsoft.2015.02.009
1364-8152/

A hydrological *model* is a computer-based tool to simulate the dynamics of a *particular* hydrological system. Example: A rainfall–runoff model of the Amazon River Basin.

A model usually comprises of two well-separated parts: (1) a generic simulation software and (2) the data describing the particular system (parameters, forcings, etc.). The first part, i. e. the plain software, is called the *model engine*. A rainfall–runoff *model engine*, for example, is applicable to many different river basins.

Finally, a *modeling framework* is a software aimed at facilitating the implementation of *model engines*. The terms *generic model (engine)* or *simulation environment* are sometimes used synonymously with *modeling framework*. The concept of modeling frameworks is introduced in the next section.

1.2. Software concepts for process-oriented modeling

In the context of water resources management, computer models proved to be quite useful tools (see, e. g. Liebscher and Mendel, 2010; Beven, 2012). This applies in particular to process-oriented, conceptional models. Such models represent a consistent collection of ideas about the functioning of real systems and they help to reveal gaps in current knowledge. Practical uses range, for example, from the assessment of management options to the generation and verification of operational forecasts.

Given the diversity of scientific and practical questions in the fields of hydrology and aquatic ecology, the water research community agrees upon the fact that a single 'all-purpose model engine' does not (and cannot) exist (Teutsch and Krüger, 2011). This is because the appropriate complexity (variables and interactions) as well as the spatio-temporal resolution of a model depends on many factors, including (1) the research question, (2) the system's characteristics, (3) the spatial and temporal scale of application, as well as (4) the available data and computational resources. Consequently, a very large number of individual model engines has been developed over the last decades (e. g. Singh and Frevert, 2002a,b; Liebscher and Mendel, 2010). They are usually specialized on the representation of selected processes or compartments.

A rather new focus of research has been set on the development of more flexible model engines. Ideally, they can be adapted to the problem of interest, for example, by considering additional processes/compartments and/or by dropping unnecessary ones. Likewise, the mathematical description of the hydrological processes, can be adapted mindful of the present knowledge, existing data, and available computer resources. The remainder of this section gives a short overview on major lines of development.

1.2.1. Multi-compartment model engines

The idea is to simulate all compartments (typically soil, subsurface, surface waters) and the related hydrological processes within a single, fully integrated software. Well known examples are HydroGeoSphere (Therrien et al., 2010; Sudicky et al., 2008), the Penn State Integrated Hydrological Model (PIHM; Qu and Duffy, 2007; Bhatt et al., 2014), Système Hydrologique Européen (SHE; Refsgaard et al., 2010) as well as the Soil and Water Assessment Tool (SWAT; Arnold et al., 1994; Arnold and Fohrer, 2005) and its derivatives. The integrated simulation of all major hydrological processes makes these tools applicable to a broad range of complex problems. A drawback, however, is the complexity of the software and the high demand for data and computation time in particular. This may effectively hamper the application in many real-world situations.

1.2.2. Component-based model engines

This concept uses largely independent model engines (components) for the simulation of individual compartments and/or processes. To allow for the necessary exchange of data between the components at run time, a communication standard is introduced. This strategy is mainly represented by the Open Modeling Interface, OpenMI (Gregersen et al., 2007) and the Framework for Risk Analysis in Multimedia Environmental Systems, FRAMES (Whelan et al., 2014) but see Dozier et al. (2014) for alternatives. The subdivision of the software into well-separated components is an evident advantage in terms of maintenance and collaborative development. The approach is, however, primarily designed for the coupling of existing model engines. It does not facilitate the development of new engines and requires profound knowledge of low-level details of both the communication standard (Castronova and Goodall, 2010) and the coupled models.

1.2.3. Modeling frameworks

A modeling framework is a software aimed at facilitating the development of different model engines. Thus, the framework is applicable to a *range of problems*. The user of the framework must define (or select) the modeled compartments including the respective state variables, forcings, and parameters. He must also specify equations to describe all processes acting on the state variables. As with component-based simulation software, framework-based model engines are well-structured. They are easily maintained, modified or extended. The framework-based approach fits well with the typical work flow where a preliminary model engine is developed rapidly with the intention of stepwise improvement. Last but not least, modeling frameworks are perfect for studying uncertainties related to the model (engine's) structure. References to a number of existing modeling frameworks can be found in Section 2.

None of the three approaches outlined above is superior in all thinkable situations. Consequently, it seems desirable to further invest in all three lines of development and to accept its coexistence.

1.3. Focus of this paper

This paper puts the focus on the third approach addressed in Section 1.2. A short survey on existing modeling frameworks is presented in Section 2 before a newly developed framework called ECHSE[2] is introduced in Section 3. It's most prominent target application is hydrological catchment modeling. Nevertheless, bearing some inherent limitations in mind (Section 3.2), the framework can be used productively in other contexts too. For example, the ECHSE could aid in the implementation of object-based water quality models for river-lake systems (see, e. g. Kneis, 2007). Likewise, the model engine used by Förster et al. (2005) to simulate flood storage in a system of interconnected reservoirs could have been implemented very easily using the ECHSE framework.

The specific objectives of software development are outlined in Section 3.1. Fundamental concepts and limitations of the framework are presented in 3.2. The subsequent sections 3.3 and 3.4 address the practical steps of model engine development and setup. An example of use is given towards the end of the paper (Section 4). Conclusions are drawn in Section 5 and information on software availability is contained in Availability of the software.

2. Existing water-related modeling frameworks

Due to compelling benefits of generic simulation tools in terms of flexibility and re-usability, a vast number of software solutions

[2] Abbreviation for Eco-Hydrological Simulation Environment.