Environmental Modelling & Software 54 (2014) 60–72

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Development of a knowledge library for automated watershed modeling

Mateja Škerjanec [a,b,*], Nataša Atanasova [c,d], Darko Čerepnalkoski [e], Sašo Džeroski [e,f,g], Boris Kompare [a]

[a] University of Ljubljana, Faculty of Civil and Geodetic Engineering, Jamova cesta 2, 1000 Ljubljana, Slovenia
[b] Water Science Institute, Hajdrihova 28a, 1000 Ljubljana, Slovenia
[c] University of Algarve, Centre for Marine and Environmental Research (CIMA), Campus de Gambelas, 8005-139 Faro, Portugal
[d] International Centre for Coastal Ecohydrology (ICCE), Solar do Capitão Mor, EN125 Horta das Figuras, 8000-518 Faro, Portugal
[e] Jožef Stefan Institute, Department of Knowledge Technologies, Jamova cesta 39, 1000 Ljubljana, Slovenia
[f] Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
[g] Centre of Excellence for Integrated Approaches in Chemistry and Biology of Proteins, Jamova cesta 39, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 12 March 2013
Received in revised form
29 November 2013
Accepted 23 December 2013
Available online 19 January 2014

Keywords:
Watershed model
Hydrology
Nutrient loading
Knowledge library
Automated modeling

ABSTRACT

In this work, we develop a library of components for building semi-distributed watershed models. The library incorporates basic modeling knowledge that allows us to adequately model different water fluxes and nutrient loadings on a watershed scale. It is written in a formalism compliant with the equation discovery tool ProBMoT, which can automatically construct watershed models from the components in the library, given a conceptual model specification and measured data. We apply the proposed modeling methodology to the Ribeira da Foupana catchment to extract a set of viable hydrological models. By specifying the conceptual model and using the knowledge library, two different hydrological models are generated. Both models are automatically calibrated against measurements and the model with the lower root mean squared error (RMSE) value is selected as an appropriate hydrological model for the selected study area.

Software availability

Name: Process-Based Modeling Tool (ProBMoT)
Developer and contact address: Darko Čerepnalkoski, Jožef Stefan Institute, Department of Knowledge Technologies, Jamova cesta 39, 1000 Ljubljana, Slovenia. Email: darko.cerepnalkoski@ijs.si
Year first available: 2012
Hardware requirements: x86 architecture, 1 GHz processor, 1 GB memory
Operating System: MS Windows (32 or 64 bit), GNU/Linux (32 or 64 bit)
Software requirements: Java 6 (32 or 64 bit)
Program language: Java
Program size: 15 MB
Availability: Available upon request from the authors

1. Introduction

Watershed modeling is recognized as a useful tool for evaluating the effects of land and water management practices on natural resources. It usually operates between various compartments, e.g., land and water, and involves many different disciplines, such as hydrology, agriculture, water management, and others. Thus, its complexity and transdisciplinary nature make it a part of the science of integrated environmental modeling (IEM, Laniak et al., 2013).

The environmental modeling community has been actively developing various watershed models, such as SWAT (Arnold and Fohrer, 2005), SPARROW (Schwarz et al., 2006) and GWLF (Haith and Shoemaker, 1987). These mostly differ in the way of conceptualizing the catchment, in the level of detail in describing catchment processes, in the specific mathematical formulations, and in the data requirements for simulation. Most models are only used by

* Corresponding author. University of Ljubljana, Faculty of Civil and Geodetic Engineering, Jamova cesta 2, 1000 Ljubljana, Slovenia. Tel.: +386 1 425 40 52; fax: +386 1 251 98 97.
E-mail addresses: mateja.skerjanec@fgg.uni-lj.si (M. Škerjanec), natasa.atanasova@icce-unesco.org (N. Atanasova), darko.cerepnalkoski@ijs.si (D. Čerepnalkoski), saso.dzeroski@ijs.si (S. Džeroski), boris.kompare@fgg.uni-lj.si (B. Kompare).

1364-8152/$ – see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.envsoft.2013.12.017

a few research groups, some gather a following within a region, and only a few become widespread.

Although it is often perceived to be easier to create a new model than to reuse an existing one, the benefits of model exchange and reuse have been largely recognized (Holzworth et al., 2010). From the user's perspective, the selection of a suitable (existing) model remains a difficult task, since various models can adequately fulfill the requirements. In other words, there is usually no single suitable model for a specific system. The model choice is guided by the modeling requirements, data availability, and modeler's skills.

To overcome this problem, a modeling concept was proposed that does not focus on searching for the best suitable model but rather on the most appropriate combination of modeling blocks (also termed modules or components) to represent the observed system (Leavesley et al., 2002; Argent, 2004). As a result larger software frameworks providing reusable components for building environmental models have been developed, e.g., the E2 (Argent et al., 2009), the Source IMS (Welsh et al., 2013) and the Object Modeling System (OMS, David et al., 2013). Such frameworks offer an infrastructure that supports inter-model communication. They usually comprise framework-dependent libraries containing reusable modules for simulating a variety of processes, where each single module represents a computable model of a part of the system. Within the selected environmental modeling framework, these executable components are coupled through the use of standard interfaces for data exchange (e.g., the OpenMI, Gregersen et al., 2007) in order to construct an integrated model of the observed system.

Libraries supporting environmental modeling frameworks usually focus on the collection and documentation of previously developed legacy models, considering each model as a single executable module. One of the most notable examples is the Library of Hydro-Ecological Modules (LHEM) introduced by Voinov et al. (2004). LHEM incorporates modules for the simulation of hydrological processes and nutrient cycling, along with other processes. Modules encoded in LHEM may be used either as stand-alone models to describe certain processes and ecosystem components, or may be put together into more complex structures by using the SME model building environment (Voinov et al., 1999).

Unlike the traditional component-based modeling approach, where each module represents a single executable model, we are introducing a new modeling approach that uses a declarative formalism for describing the system. Although we also decompose the modeling domain into several components, none of them represent an executable model by itself; rather, they represent the entities and processes involved in the domain of study. While entities correspond to the actors of the observed system, processes are used to define the relationships among them. Within the automated model generation procedure, all components are compiled together in order to produce a global model that encompasses all parts of the system.

In this work, we present (1) a domain-specific library that contains formalized watershed modeling knowledge and (2) a case study demonstrating the utility of the developed library for the extraction of viable watershed-scale hydrological models. At present, the knowledge in the library comprises hydrological processes, based on meteorological data, and nutrient loading processes, considering point and diffuse emission sources. Moreover, the library includes alternative formulations for the selected processes.

The library can be considered as an ontology, because it consists of organized and structured modeling knowledge. Similarly to other ontologies, it defines concepts (i.e., entities and processes) and the relationships between them. The taxonomies of entities and processes provide the inheritance (is-a) relation that is the essential part of an ontology. However, our domain library is much richer than the typical ontology. Besides listing the concepts and taxonomical relations between them, our library contains their properties (i.e., variables and constants for entities and equations for processes). The latter are the basic components that are put together to construct dynamical models in the form of systems of differential and/or algebraic equations.

The library can be used as a repository of modeling components when handcrafting semi-distributed watershed models running at a daily time step. The true usefulness of the library comes from its use by an automated modeling tool, such as Lagramge (Džeroski and Todorovski, 2003), HIPM (Todorovski et al., 2005), or the recently developed ProBMoT (Čerepnalkoski et al., 2012). These tools allow automatic induction of suitable models based on the libraries of domain-specific modeling knowledge and the measured data.

By developing a watershed library compliant with the ProBMoT, we are establishing a novel approach to automated modeling (AM) of watersheds that uses a combination of theoretical and data driven modeling. A similar approach, based on an aquatic ecosystem library (Atanasova et al., 2006), was successfully applied for lake food web modeling (Atanasova et al., 2011, 2008). However, to our knowledge, no such attempts have been made at a watershed scale, despite the similarity between the two modeling problems.

The paper is organized as follows. First, in Section 2, we briefly explain the watershed modeling domain. Next, in Section 3, we present the formalism for encoding the watershed modeling knowledge into the library and explain how the modeling task specification is included in the model induction procedure. In the following section (Section 4) we introduce ProBMoT. In Section 5, we present the model generation, calibration and validation for an experimental watershed. This is followed by a discussion (Section 6). Finally, conclusions and guidelines for further work are given in Section 7.

2. Watershed processes and modeling

When simulating the loadings of water quality constituents (sediment and nutrients) from watersheds, two basic groups of processes have to be taken into account: hydrological processes and constituent generation processes from various land use types, triggered by the water movement. The difference between precipitation and water losses (evapotranspiration, infiltration, percolation) results in the surface runoff and subsurface (groundwater) discharge. The generated surface and groundwater flows provoke soil erosion and constituent wash-off to various surface water recipients and ground water reservoirs. Besides the natural water cycle and related constituent loadings, we also have to consider human-generated water flows, such as septic effluents and other point sources (waste water treatment plants – WWTPs, industry discharges, and others) rich with nutrients.

Existing watershed models simulate these processes at different levels of detail. Physically based dynamic models (e.g., SWAT, Arnold and Fohrer, 2005; HSPF, Donigian et al., 1995) have a highly complex mass-balance structure and provide the best representation of the current understanding of watershed processes affecting pollution generation. However, the parameterization and calibration of this kind of models can be very difficult and time consuming. In contrast, the predominantly empirical steady state models (e.g., SPARROW, Schwarz et al., 2006; MONERIS, Behrendt et al., 1999) are compilations of expert knowledge and empirical relationships between the physiographic characteristics of the watershed and constituent loadings. Empirical models are conceptually simple and tend to be less expensive to implement compared to more physically based approaches. They commonly

provide an excellent fit to the observations (if properly calibrated), but have the disadvantage of providing little understanding of the watershed processes. If calibration cannot be performed (e.g., due to lack of data) the use of empirical models may not be appropriate.

Other modeling methods can be placed between the two above mentioned extremes. These so called "mid-range" methods present a compromise between simple empirical export coefficients (that predict annual losses of nutrients to the water) and complex simulation models (that require large amounts of detailed data (US EPA, 1999)). A representative of the class of "mid-range" methods is GWLF (Haith and Shoemaker, 1987).

Many of the above mentioned watershed models use semi-distributed modeling approach based on spatial discretization into subcatchments and functional units. In this case, hydrological and constituent generation processes are simulated for each functional unit. This kind of discretization introduces some spatial variability, though it does not explicitly consider the geographical locations of the functional units in the subcatchment.

In this work, we develop a knowledge library for the domain of watershed modeling. We aim to integrate the essentials of various existing watershed models under the same umbrella. For the initial setup of the library, we used a watershed modeling concept similar to the one introduced by Haith and Shoemaker (1987), because it offers an acceptable level of complexity, taking into account all the basic watershed processes. Please note that the library is developed in a flexible manner, which means that it can be easily extended with additional processes and alternative formulations for each process.

The processes we have encoded in our watershed modeling library are presented in Fig. 1 and are briefly explained below.

2.1. Hydrological processes

2.1.1. Surface runoff

For runoff prediction, we used the widely applied and well documented Soil Conservation Service (SCS) curve number (CN) method considering the effect of antecedent moisture conditions. The method includes the determination of the dry condition CN and the wet condition CN, both calculated based on the average condition CN. The typical values for the latter are listed in the tables provided by the SCS Engineering Soil Conservation Service Engineering Division (1986).

2.1.2. Evapotranspiration

Evapotranspiration is calculated as the product of the potential evapotranspiration (PET) and a cover coefficient (CV; Wu et al., 2010) and is limited by available moisture in the unsaturated zone. For the PET calculation (mm) we selected two alternative formulations:

$$PET_1 = 0.21 \cdot H^2 \cdot e/(T + 273) \tag{1}$$

$$PET_2 = 0.0023 \cdot H_0 \cdot (T_{max} - T_{min})^{0.5} \cdot (T_{avg} + 17.8)/\lambda \tag{2}$$

In Eq. (1) (Hamon, 1961), H is the number of daylight hours per day, e is the saturated water vapor pressure (mbar), and T is the temperature on a given day (°C). In Eq. (2) (Hargreaves et al., 1985), H_0 is the solar radiation (MJ m^{-2}), T_{max}, T_{min}, T_{avg} are the maximum, minimum and mean air temperatures for a given day (°C), and λ is the latent heat of vaporization (MJ kg^{-1}).

2.1.3. Subsurface water movement

Subsurface water fluxes are determined by using a water balance approach. The daily water balances for the unsaturated and shallow saturated zones are as follows:

$$dU/dt = Prec - Q - ET - Perc \tag{3}$$

$$dSS/dt = Perc - GW - DS \tag{4}$$

where U is the unsaturated zone water (mm), $Prec$ is precipitation (mm), Q is the amount of water available for surface runoff (mm), ET is evapotranspirated water (mm), $Perc$ is percolated water (mm), SS is shallow saturated zone water (mm), GW is the amount of generated groundwater (mm) and DS is the amount of water loss due to deep seepage (mm).

Percolation occurs only when the unsaturated zone water exceeds the available soil water capacity (U^*, in mm) and is calculated as:

$$Perc = \max\left(0;\ U + Prec - Q - ET - U^*\right) \tag{5}$$

Groundwater discharge (GW, in mm) is calculated as the product of the shallow saturated zone water (SS, in mm) and the groundwater recession constant (r). A similar formulation is applied for the calculation of the deep seepage (DS, in mm), using a different (seepage) constant (s).

$$GW = SS \cdot r \tag{6}$$

$$DS = SS \cdot s \tag{7}$$

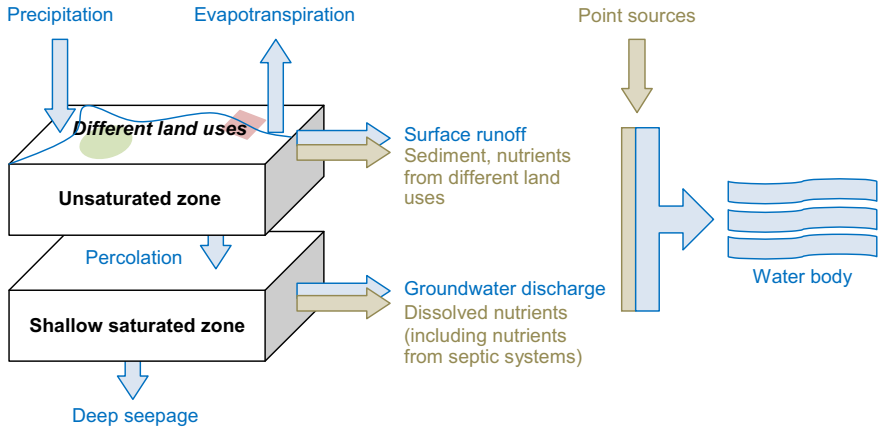


Fig. 1. Basic processes currently encoded in the watershed modeling library. Blue arrows represent hydrological processes, while brown arrows symbolize constituent loadings. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2. Constituent generation processes

2.2.1. Sediment yield

Sediment yield (*sed*, in metric tons) can be calculated by selecting one of the two alternative formulations:

$$sed_1 = 0.132 \cdot RE \cdot Q^{5/3} \cdot A \cdot K \cdot C \cdot AP \cdot LS \cdot DR \quad (8)$$

$$sed_2 = 11.8 \cdot \left(Q \cdot q_{peak} \cdot A\right)^{0.56} \cdot K \cdot C \cdot AP \cdot LS \quad (9)$$

Eq. (8) was initially introduced by Haith (1985) and is based on the Universal Soil Loss Equation (USLE, Wischmeier and Smith, 1978). In Eq. (8), 0.132 is a dimensional conversion factor, *RE* is the rainfall erosivity on a given day, *Q* is the amount of water available for surface runoff (mm), *A* is the source area (ha), and *DR* is the sediment delivery ratio. Eq. (9) demonstrates the Modified Universal Soil Loss Equation (MUSLE, Williams, 1995), where *Q* is the surface runoff volume (mm/ha), q_{peak} is the peak runoff rate (m^3/s), and *A* is the source area (ha).

K, *C*, *AP* and *LS* are the standard USLE factors and are the same for both equations. *K* is the soil erodibility factor (t h MJ^{-1} mm^{-1}), *C* is the cover factor, *AP* is the agricultural practice factor, and *LS* is the topographic factor.

2.2.2. Dissolved nutrient loadings

Dissolved nutrient loadings are calculated separately for nitrogen (N) and phosphorous (P) by summing dissolved loadings from point sources, surface runoff, groundwater and septic systems. Dissolved surface runoff loadings (L_{sr_D}, in kg) are calculated based on the average nutrient concentrations in surface runoff (C_{sr}, in mg L^{-1}):

$$L_{sr\_D} = 0.01 \cdot C_{sr} \cdot Q \cdot A \quad (10)$$

A similar approach is used for the calculation of dissolved groundwater loadings. Dissolved nutrient loadings from septic systems (L_{sept_D}, in kg) are estimated by using the per capita daily loads from each septic system (e_N, in g $capita^{-1}$) reduced by plant uptake (u_N, in g $capita^{-1}$), multiplied by the number of inhabitants served by such a system (n_{inh}) and the portion of annual groundwater discharge generated in a single day (GW/GW_{ann}, both in mm):

$$L_{sept\_D} = 0.001 \cdot n_{inh} \cdot (e_N - u_N) \cdot GW/GW_{ann} \quad (11)$$

2.2.3. Solid nutrient loadings

Solid-phase nutrient loadings are calculated separately for N and P by summing solid-phase loadings from rural and urban surfaces. Rural solid-phase loadings are calculated for each non-urban land use type as the product of average soil-nutrient concentration, an enrichment ratio and sediment yield. Urban solid-phase loadings (L_{urb_S}, in kg) can be calculated by using either the exponential accumulation and wash-off function (Eq. (12)) or the USGS regression equation (Eq. (13)).

$$L_{urb\_S1} = \left(1 - e^{-1.81 \cdot Q/10}\right) \cdot N \cdot A \quad (12)$$

$$L_{urb\_S2} = \beta 0 \cdot (Prec/25.4)^{\beta 1} \cdot (A \cdot imp/2.59)^{\beta 2} \cdot (imp \cdot 100 + 1)^{\beta 3} \cdot \beta 4/2.205 \quad (13)$$

In Eq. (12) (Haith et al., 1992), *Q* is the amount of water available for surface runoff (mm), *N* is the accumulated nutrient loading (kg ha^{-1}), and *A* is the source area (ha). In Eq. (13) (Driver and Tasker, 1988), *Prec* is the precipitation on a given day (mm), *A* is the drainage area (km^2), *imp* is the fraction of the area that is impervious, and the β variables are regression coefficients.

3. Encoding watershed modeling knowledge

The watershed modeling knowledge presented in Section 2 has been encoded into a domain-specific library (Appendix 1) which can be used by ProBMoT (Čerepnalkoski et al., 2012). The library employs a process-based formalism using domain specific language (DSL) principles. The formalism provides means of describing the components included in the library in a formal and precise manner. Each component is specified as one data structure, which has its own unique name and a set of properties.

Process-based models contain two types of components: entities and processes. Entities represent the actors of the observed system. These actors are involved in processes that explain how entities interact, as well as what is the influence of the interactions on the involved entities themselves. When we deal with equation-based models, entities correspond to the variables in the equations and processes to arithmetical expressions (equation fragments).

In the watershed modeling domain, entities correspond to different pools within the water cycle, climate variables and various types of constituents (see Fig. 2A). Each entity is uniquely identified by its name. It is also described with one or more properties that are fixed, called constants, and properties that can change with time, called variables. Processes provide quantitative descriptions of the relations they represent as one or more equations. Furthermore, an equation can contain only variables and constants of the entities that participate in the corresponding process. In the watershed modeling domain, examples of processes include water fluxes, i.e., transfer processes that are involved in water cycle, and constituent loadings (see Fig. 2B).

The dynamic system we want to model, i.e., the watershed, can be structured by using compartments. Compartments are organized in a tree-like structure. Each compartment contains entities and processes and can also contain other sub-compartments (e.g., subcatchments or functional units).

The process-based modeling formalism explained above uses two-phase specification. The general modeling knowledge about compartments, entities, and processes is specified in the library. The library consists of entity templates, process templates, and compartment templates. Each template captures general knowledge that applies to different cases and can be reused when dealing with a specific task. In a particular process-based model which is intended to describe a certain use-case, the templates from the library are instantiated to obtain entity instance, process instances, and compartment instances. Each instance is created from a single template and acquires all properties of that template. Furthermore, each instance can provide additional details pertinent to the particular system being modeled.

3.1. Watershed modeling library

As mentioned earlier, the watershed modeling library consists of compartment templates, entity templates, and process templates. Their specification is presented in the following paragraphs. Please note that we used our own notation for naming particular templates.

A *compartment template* specifies the content of a compartment. Compartments are containers for structuring a system. A system can contain one or more compartments, or none at all, and each compartment can contain other compartments as sub-compartments. A compartment contains entities and processes, which is its primary function. Hence, compartments form a tree-

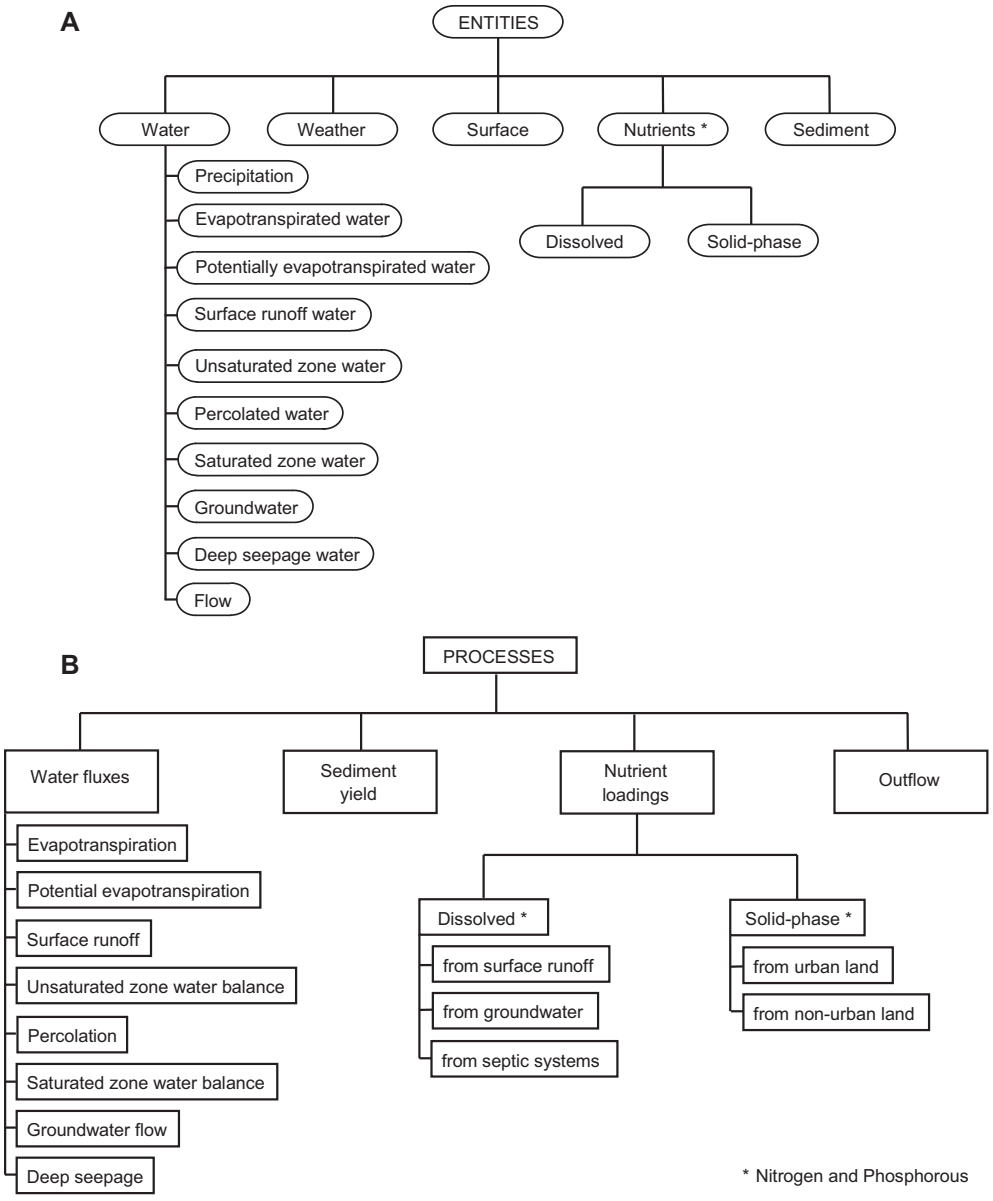


Fig. 2. Hierarchical organization of A) entities and B) processes encoded in the watershed modeling library.

shaped hierarchy with the whole system at the top, which itself can be considered as a top-level compartment. The specification of a compartment template consists of the name of the compartment and a list of entity, process, and compartment templates which can be contained within the compartment. Fig. 3 shows the specification of the compartment template "SpatialUnit" which is the most generic compartment encoded in the watershed modeling library.

```
template compartment SpatialUnit {
  entities:
      Water,
      EvapotranspiratedWater,
      PotentiallyEvapotranspiratedWater,
      Weather,
      Surface,
      ...;
  processes:
      Evapotranspiration,
      PotentialEvapotranspiration,
      ...;
  compartments:
      SpatialUnit;
}
```

Fig. 3. Specification of the compartment template – an example.

An *entity template* defines the basic structure of entity instances. Its specification consists of a name, a list of variables and a list of constants.

Variables (Appendix 2) and constants (Appendix 3) are very similar in structure. The crucial difference between them is that variables represent properties that change over time, whereas constants are properties whose values remain fixed. The specification of a constant includes its name, range of allowed values, and the unit of measurement. Similarly, the specification of a variable also includes its name, measurement unit and range. In addition, a variable also includes an aggregation function, which can be sum, product, minimum, maximum, or average. The role of the aggregation function comes into play when several processes influence the same variable. Thus, all influences, represented as equation fragments, are combined into a single equation with the provided aggregation function.

```
A) template entity Water {
     vars:
         amountOfWater {aggregation:sum; unit:"mm/day"; range:<0,inf>};
   }

B) template entity EvapotranspiratedWater : Water {}

C) template entity Surface {
     consts:
         area {range:<0,inf>; unit:"ha"};
   }
```

Fig. 4. Specification of entity templates. Examples A and B are in a hierarchical relationship. The entity template in example A defines a parent entity with a list of variables, while example B defines its sub-entity. Example C shows the specification of an entity template that contains constants only. Hierarchical organization of A) entities and B) processes encoded in the watershed modeling library.

Entity templates can be arranged into inheritance trees. The more general properties are placed in the entity templates which are higher up the tree. This enables the entity templates which are lower in the tree to inherit the properties of their ancestors and provides a modular, clean, and reusable design of the entity templates. Once we specify an attribute (variable or constant) in the entity template, its specification remains the same for all of its sub-entities. In other words, none of the sub-entities can redefine the specified attributes. For each template entity, we include the name of its parent entity directly in its specification.

In Fig. 4A, we give the specification of the template entity "Water", which can be described by a single variable "amountOfWater". Its unit of measurement is mm/day and it can take any value between 0 and infinity. If the values of the variable "amountOfWater" are to be aggregated, they are to be summed. Fig. 4B shows the specification of the template sub-entity "EvapotranspiratedWater", whose parent is the template entity "Water". Because the sub-entities inherit all the attributes from the parent, the entity "EvapotranspiratedWater" can also be described with the variable "amountOfWater". Fig. 4C represents the specification of the template entity "Surface", which can be described by a single constant "area". Its unit of measurement is ha and it can take any value between 0 and infinity.

A *process template* specifies a recipe for creating process instances. Its specification consists of a name, a list of arguments (the entities involved), its parent process and its inner specification. The inner specification contains the constants and the equations that are part of the process. These equations can include references to the arguments' variables and constants as well as references to the processes' constants.

Fig. 5A presents an example of a template process specification, namely the specification of the template process "Evapotranspiration". The entities involved in this template process and their abbreviations are given in parentheses, i.e., "et : EvapotranspiratedWater" and "pet : PotentiallyEvapotranspiratedWater", etc. There is only one constant that takes part in this process ("coverCoeff"), which can take any value between 0.1 and 1.6. The process "Evapotranspiration" can be described with a single equation, where the value of the entity "EvapotranspiratedWater", namely its variable "et.amountOfWater", is to be calculated.

Process templates are also organized into taxonomy. Each process template inherits the properties of its ancestors, such as the arguments of the process and the contained equations. The taxonomy of process templates serves one very important purpose. The process templates that are at the higher levels of the taxonomy represent conceptual processes, which are not bound to any particular equation. On the other hand, process templates which are at the bottom of the taxonomy represent specific mathematical formulations of the conceptual processes. Thus, the process template taxonomy enables the domain expert to represent alternative process formulations.

The hierarchy of template processes in the watershed modeling library is represented in Fig. 5B–D. The template process "PotentialEvapotranspiration" (Fig. 5B) is a parent (or a super-process) of the processes "PotentialEvapotranspirationHamon" (Fig. 5C) and "PotentialEvapotranspirationHargreaves" (Fig. 5D). At the same

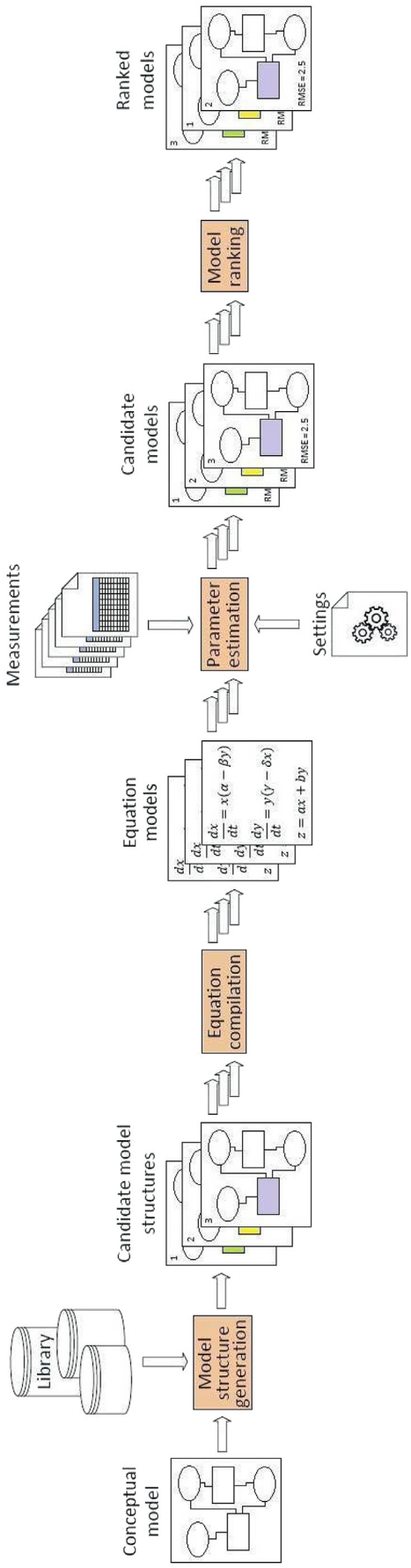
```
A) template process Evapotranspiration (et : EvapotranspiratedWater, pet
: PotentiallyEvapotranspiratedWater) {
     consts:
         coverCoeff {range:<0.1,1.6>};
     equations:
         et.amountOfWater = coverCoeff * pet.amountOfWater;
   }

B) template process PotentialEvapotranspiration (pet :
   PotentiallyEvapotranspiratedWater, w : Weather) {
     consts:
         lambda {range : <0,2.257>; unit : "MJ/kg"};
   }

C) template process PotentialEvapotranspirationHamon :
   PotentialEvapotranspiration {
     equations:
         pet.amountOfWater = (0.21 * pow(w.daylightHours,2) *
         w.satVapPressure) / (w.temperature + 273);
   }

D) template process PotentialEvapotranspirationHargreaves :
   PotentialEvapotranspiration {
     equations:
         pet.amountOfWater = (0.0023 * w.radiation * pow(w.maxTemperature
         - w.minTemperature,0.5) * (w.temperature + 17.8)) / lambda;
   }
```

Fig. 5. Specification of process templates. Example A presents a basic process template specification with a list of arguments (in brackets), a list of constants, and a list of equations. Example B shows the definition of a super-process (a parent), in the process hierarchy, while examples C and D define two alternative sub-processes of this.



time, the latter two processes represent the two alternative formulations of the process "PotentialEvapotranspiration", i.e., the alternative formulations for the calculation of the variable "pet.amountOfWater".

3.2. Conceptual model specification

In order to apply ProBMoT to a specific watershed, we have to provide a conceptual model of the observed system. The conceptual model consists of entity, process, and compartment instances. Each instance is created using one template from the watershed modeling library.

The specification of each instance begins with a unique name and a property indicating the template to which a particular instance belongs. When specifying *entity instance*, we have to add a list of variables together with their initial values and their role in the model (either endogenous/dependent or exogenous/independent). Where applicable, we also have to provide a list of constants with their numeric values. The specification of a *process instance* has to include a list of arguments (i.e., names of the entity instances that are involved in the process) and a list of constants with their exact values. In case we do not know the exact values of the constant parameters or we want to calibrate them, they are assigned a special value *null*.

4. ProBMoT: a process-based modeling tool

4.1. Structure and functioning

The proposed automated modeling (AM) approach (Fig. 6) is based on the process based modeling tool ProBMoT, developed by Čerepnalkoski et al. (2012). ProBMoT takes into account domain-specific knowledge formalized as templates for the components of the process-based models. It automatically identifies both the structure and parameter values of the appropriate process-based model, given: a) a conceptual model of the observed system, b) the library of domain knowledge, and c) the measurements.

In the first stage of the proposed AM procedure, using the components from the library, ProBMoT generates all candidate model structures that adhere to the conceptual model specified as input. Next, each model structure is translated into a set of algebraic and/or ordinary differential equations. For each spatial unit (i.e. compartment), it assembles a set of equations that represent all process influences in that compartment. In addition to these, there are equations for processes which represent inter-compartmental relations. The collection of all equations for all compartments and all inter-compartmental processes represents the model of the whole system. This model is valid at all time points. For each state variable in the model, ProBMoT builds one equation, which has that state variable as the left-hand-side of the equation. On the right-hand-side, ProBMoT assembles all equations from all processes that influence that variable (we denote these as equation fragments). It combines the equation fragments with an aggregation function which is specified in the definition of the state variable (where no aggregation function is specified, summation is assumed by default).

Fig. 6. A schematic description of the functioning of ProBMoT. Candidate model structures are generated from the modeling library and a user-specified conceptual model of an observed system. The candidate models are transformed into equations, calibrated against measurements and ranked according to their RMSE values. Ovals in schematic representations of models depict entities while rectangles stand for processes. Colored rectangles represent different process formulations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

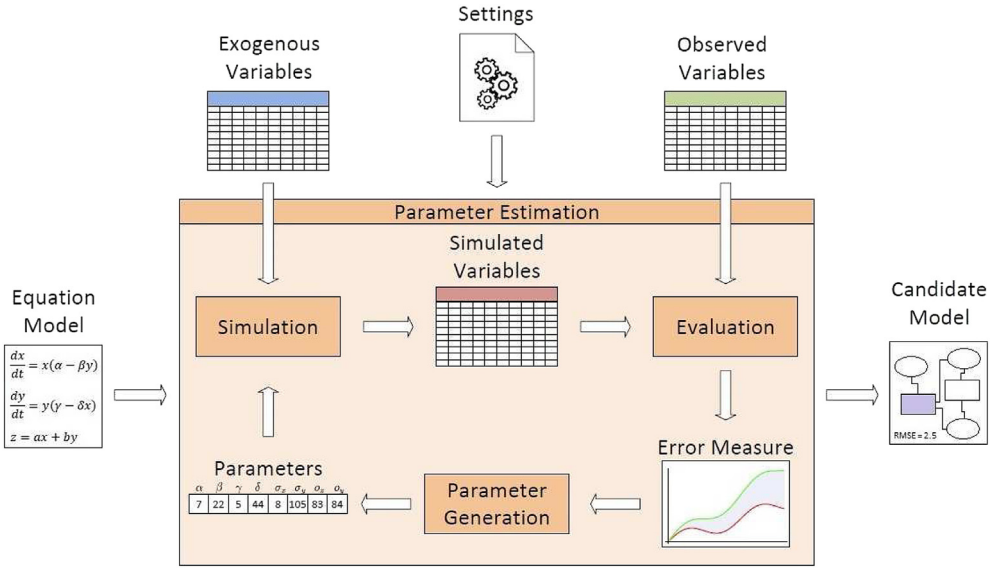


Fig. 7. A detailed overview of the parameter estimation phase of ProBMoT.

The resulting set of equations (internally represented as a Java method) is entering the parameter estimation subsystem (Fig. 7). The Java method takes as input an array of time points at which the system's behavior is observed, and one array of measurements for each exogenous variable that appears in the model. For each state (i.e. endogenous) variable, it outputs one array with the simulations of that state variable at each time point given as input. The parameter estimation subsystem uses the discrepancy between the simulated endogenous and the corresponding observed variables to find suitable values for the constant parameters of a given model. ProBMoT supports the use of different objective functions for measuring the above mentioned discrepancy, among which we use the root mean squared error (RMSE). We need measured data for at least one endogenous variable in order to perform the calibration of multiple calibration endpoints, but we can use measurements for more than one, in which case the RMSE values for each endogenous variable are normalized and aggregated. The linkage (mapping) between the variables used by the model and their measured values is specified within the settings file which also contains information about the desired optimization method. In our case, we selected a non-linear metaheuristic optimization method called Differential Evolution (Storn and Price, 1997).

The result of the parameter estimation process is a list of candidate models with fully specified structure and parameter values. These are further ranked according to their RMSE values. In this work, we are only interested in the top ranked model. However, in principle, other highly ranked models may be considered, possibly by taking into account additional quality criteria (other than RMSE).

4.2. Specification of complete and incomplete conceptual models in ProBMoT

In the theoretical modeling approach, a conceptual model represents a completely determined system with fully specified system variables and processes that relate these variables. In contrast, ProBMoT supports both complete and incomplete conceptual models of observed systems. This level of the conceptual model specification directly influences the number of candidate mathematical models to be later optimized against measured data. The more the conceptual model structure is defined, the smaller the number of candidate models.

In a *complete conceptual model*, all entities and processes are known and specified. However, each of the processes in the conceptual model can have different mathematical formulations, resulting in various mathematical models generated from a single complete conceptual model (Fig. 8). In this case, ProBMoT first enumerates all possible structures of the system, i.e., all combinations of alternative process formulations. Then, it calibrates each model structure in order to discover the best parameter values of

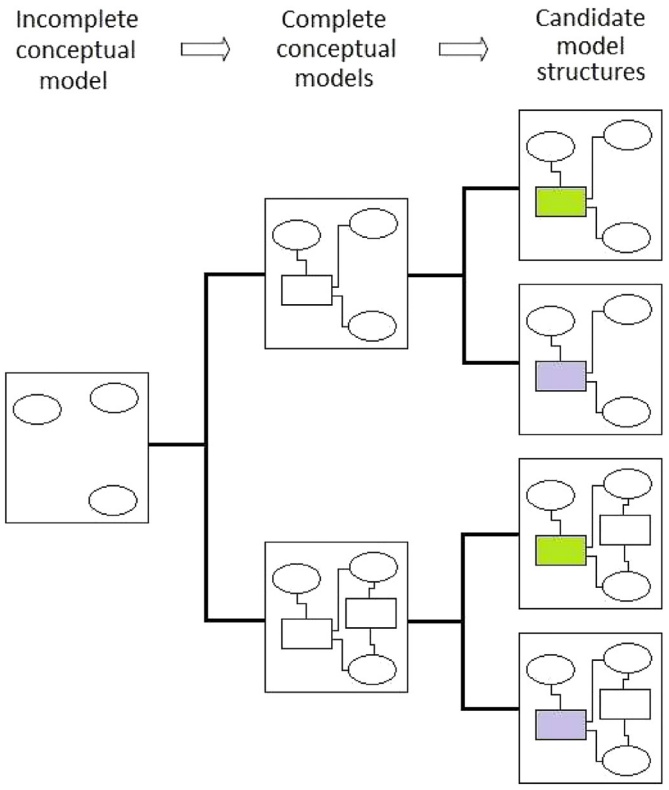


Fig. 8. Generation of candidate model structures by using either a complete or an incomplete conceptual model. Ovals represent entities while the rectangles depict processes. Colored rectangles represent different process formulations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

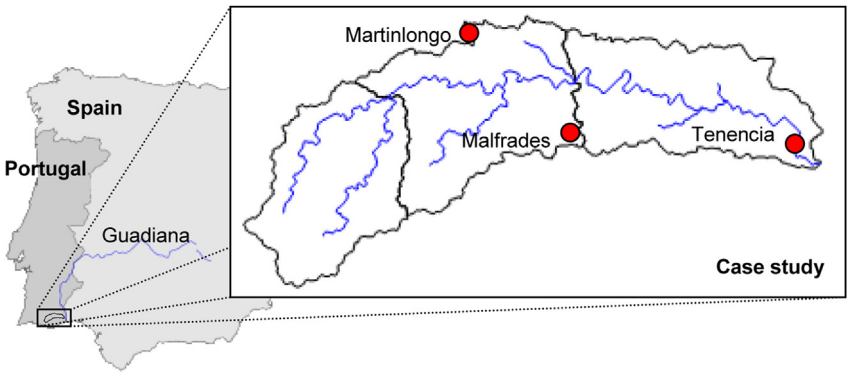


Fig. 9. The experimental subcatchments located in the Foupana River catchment in the south of Portugal.

each model. The model with the lowest RMSE is the overall best model.

In addition, the conceptual model can contain processes with a completely specified mathematical formulation. In the case where all processes in the model are specific processes, the search space of candidate models is reduced to a single model structure. Thus, the only remaining task is the parameter calibration. Furthermore, if the user provides the exact parameter values, ProBMoT can skip both structure and parameter identification and proceed directly to model simulation.

An *incomplete conceptual model* can represent multiple conceptualizations of the system, each contributing to the generation of at least one candidate model structure (Fig. 8). Conceptual model can be incomplete in two ways. In the first way, a single general process can be selected which can be described with various process conceptualizations, i.e., with different alternative (sub)processes of varying complexity. In the second way, an "empty" process can be included among the alternatives, indicating that the process doesn't exist. In both cases, the AM tool needs to perform structure identification and parameter calibration.

5. Case study

This case study demonstrates the use of the proposed AM framework for the automatic generation of semi-distributed hydrological models. The study area is located in the eastern part of the Algarve region, southern Portugal (see Fig. 9). It includes the entire catchment of the Foupana River, a tributary of the Guadiana River, with an area of 411 km^2. The study area includes three experimental subcatchments, each involving several land uses and various soil types.

5.1. Conceptual model setup

The conceptual model of the selected watershed (Appendix 4) was specified in a way that allows the generation of semi-distributed hydrological models from the library. When elaborating the conceptual model of the study area, the selected watershed was presented as a single super-compartment "watershed", divided into three compartments, i.e., "sub1", "sub2" and "sub3" corresponding to subcatchments 1, 2 and 3. These were further subdivided into sub-compartments representing different functional units characterized by homogeneous land use. All the above mentioned compartments actually represented the instances of the compartment template "SpatialUnit", encoded in the watershed modeling library.

For demonstration purposes we decided to aggregate different land use types, determined on the basis of the Corine land cover map, into three functional units, namely artificial, agricultural and natural areas. This resulted in two functional units for subcatchments 1 and 3 and three functional units for subcatchment 2 (seven altogether, see Fig. 10). The selected functional units were also quite

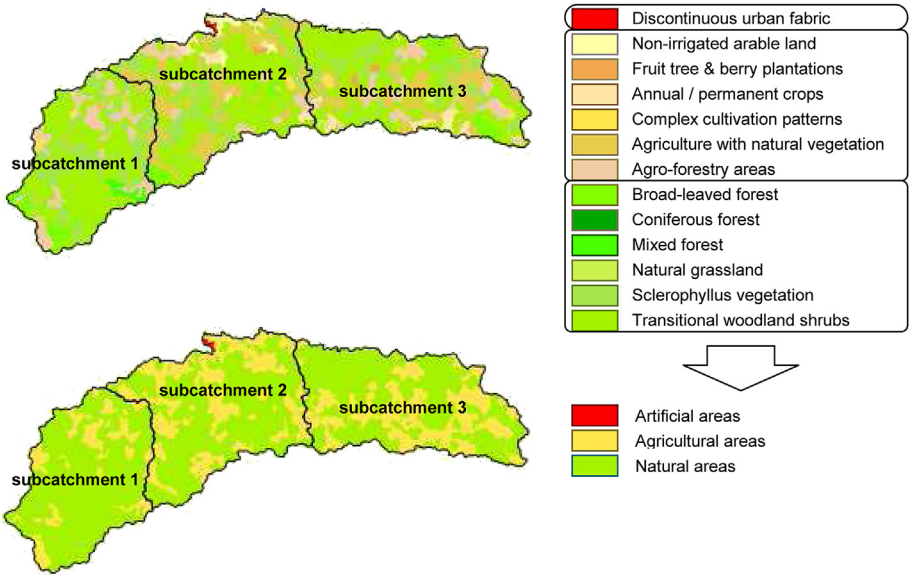


Fig. 10. The formation of functional units for the selected case study based on the aggregation of different land use types.

homogeneous with respect to their soil characteristics, derived from a regional soil map. Any further division of subcatchments would largely increase the size of the conceptual model code.

For each (sub)compartment we defined entities with their dependent (state) and independent variables. We also specified processes that represent interactions between the involved entities, namely their variables. For each sub-compartment, corresponding to a single functional unit, we considered all the hydrological processes, presented in Section 2.1.

5.2. Generation of models

Given the conceptual model (Appendix 4) and the developed library (Appendix 1), the search algorithm of ProBMoT generated 128 hydrological models for the selected study area. This number of models was obtained because each of the seven sub-compartments, corresponding to the selected functional units, could use either of the two alternative formulations for the calculation of PET, presented in Section 2.1.2. Consequently, a number of alternatives (i.e., two) had to be raised to the number of sub-compartments (i.e., seven), resulting in 128 different candidate models.

Since calibrating 128 models is computationally very expensive, we decided to further limit the search space of candidate model structures by introducing an additional constraint to the library, allowing the use of only one ET model structure for the entire experimental catchment. Adding this constraint resulted in just two candidate models, namely one using the Hamon PET equation and the other using the Hargreaves PET equation in all seven sub-compartments.

5.3. Optimization and selection of the best model

To start with the optimization (calibration) phase of ProBMoT, we first had to prepare input data. For each sub-compartment, we had to provide the daily values of the following exogenous variables: precipitation, minimum, maximum and average temperature, solar radiation, daylight hours and saturated vapor pressure. The above mentioned data were obtained from the SNIRH (Sistema Nacional de Informação de Recursos Hídricos) database for the meteorological stations Malfrades and Martim Longo (see Fig. 9) and for the period of January 1, 1998 to March 31, 1998. During the optimization phase, the selected parameters (curve number, cover coefficient, available soil water capacity, groundwater recession constant and seepage constant) were automatically calibrated against measurements based on a comparison of the calculated outflows from the selected study area to the flows measured at the hydrological station Tenencia (see Fig. 9).

Calibration of the two models took three hours of computational time (using Intel Core i7-2600K CPU (3.4 GHz) with 8 GB RAM). For each model and for each sub-compartment, we obtained values of the selected parameters. Furthermore, the root mean squared error (RMSE) was generated for each of the two models. The model with the lower RMSE value was selected as the best hydrological model for the selected study area. As it turned out, the model that used the Hargreaves equation for the calculation of PET (RMSE 2.466) performed slightly better than the model that used the Hamon equation (RMSE 2.508). Fig. 11 shows the simulation results of the better model, providing a comparison between the measured and simulated outflows from the selected study area.

The generated hydrological model allows for the analysis of the contribution to the river flow per each subcatchment and each functional unit. According to Fig. 12A, subcatchment 2 contributes 45% of the total amount of water, while subcatchments 1 and 3 contribute 28 and 27%, respectively. If we take a closer look at the

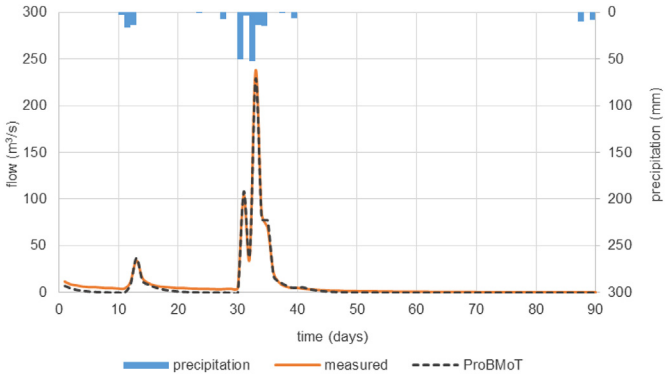


Fig. 11. Model calibration results: a comparison between the measured and simulated outflows at the Tenencia hydrological station.

subcatchment 2 (Fig. 12B), we can see that the artificial areas contribute a bit more water (38%) than the other two functional units (each contributing 31%). Further decomposition of contributions to the river flow indicates the prevailing contribution of surface runoff as compared to groundwater discharge (Fig. 12C).

5.4. Model validation and performance analysis

The selected model was validated on a second independent set of observations. For this purpose, we used the data for the meteorological station Martim Longo (see Fig. 9), for the period of January 1, 1999 to March 31, 1999. Fig. 13 shows the model validation results, namely the comparison between the simulated outflows and the flows measured at the hydrological station Tenencia (see Fig. 9).

Table 1 summarizes the values of the following model evaluation statistics: the coefficient of determination (R^2), the Nash–Sutcliffe coefficient (NSE, Nash and Sutcliffe, 1970), the Percent bias (PBIAS, Gupta et al., 1999), and the RMSE-observations standard deviation ratio (RSR, Moriasi et al., 2007). Values of the above mentioned statistics are given for both the calibration and the validation periods. Based on the model evaluation guidelines, proposed by Moriasi et al. (2007), the values of almost all statistical parameters indicate a very good model performance for both modeling phases. The only exception is the PBIAS coefficient, which measures the average tendency of the simulated data to be larger or smaller than the corresponding observations. The PBIAS value for the calibration phase indicates a tendency to underestimate the measurements. For the validation phase, its value falls below zero indicating an overestimation bias.

6. Discussion

6.1. Summary and outlook

In this paper, we introduced an automated modeling (AM) approach allowing automated induction of semi-distributed watershed models from measured data. The key step in this process is the encoding of existing watershed modeling knowledge into a domain specific library. Given the specification of the observed system, i.e., a conceptual model, ProBMoT transforms the conceptual model into different model structures for the observed system (where each structure represents one concretization of the conceptual model). The parameters in these structures are later optimized according to given measurements for the specific watershed under study.

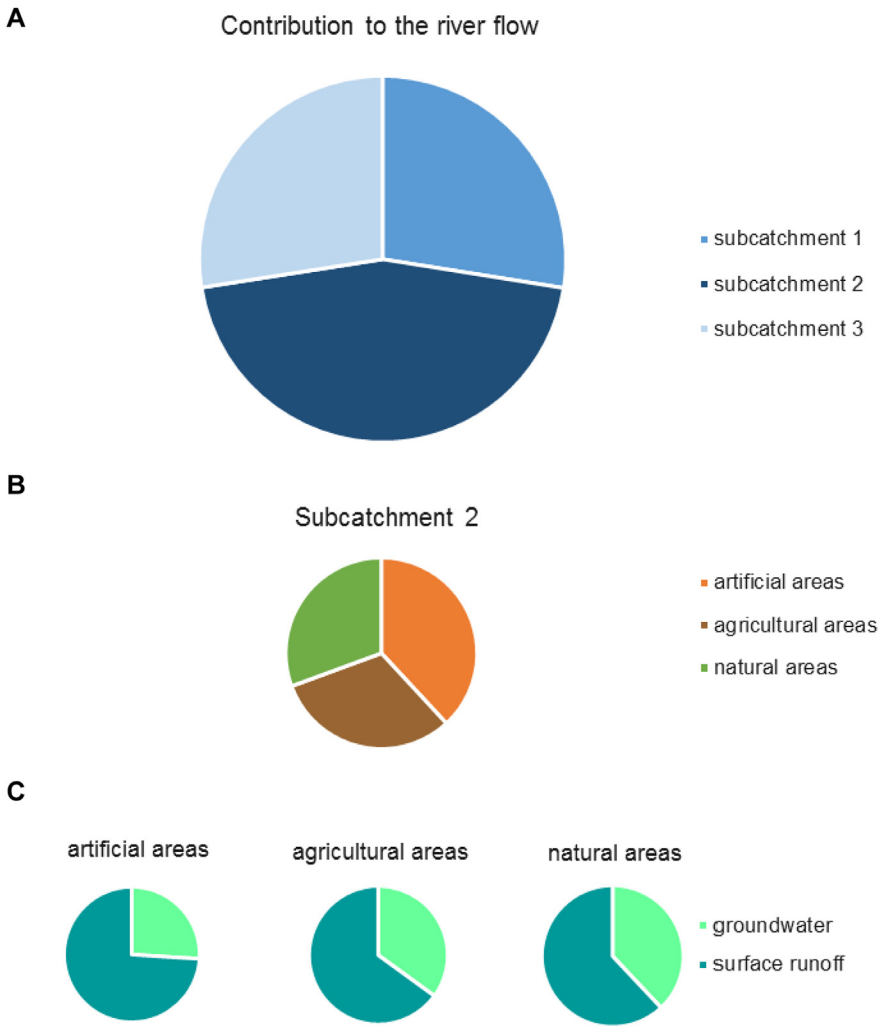


Fig. 12. Model based analysis of the contributions to the river flow: A) for the whole catchment, B) for subcatchment 2, and C) for all three functional units of subcatchment 2.

The formalism selected to build the watershed modeling library is compliant with the AM tool ProBMoT (Čerepnalkoski et al., 2012). The main advantage of this formalism is that it allows precise structuring of the watershed modeling knowledge by decomposing it into elementary modeling components, namely entities and processes. While entities correspond to particular actors of the observed system, processes are used to describe the relations among them. The meaning of each single component, its properties and accompanying logical constraints are defined through the use of metadata. Such formalization of modeling knowledge allows an AM tool to search through the space of all possible combinations of components.

A similar formalism supported by the AM tool Lagramge (Džeroski and Todorovski, 2003) was used for constructing an aquatic ecosystems modeling library (Atanasova et al., 2006) and was successfully used for several ecological modeling tasks (Atanasova et al., 2008, 2011). Like the ProBMoT formalism, the Lagramge formalism also provides hierarchical knowledge representation. However, ProBMoT has a number of advantages as compared to Lagramge, discussed below.

ProBMoT uses a hierarchy of entities instead of a hierarchy of variables. Thus, it enables new features that are not supported by Lagramge, such as metadata specification. Using the ProBMoT

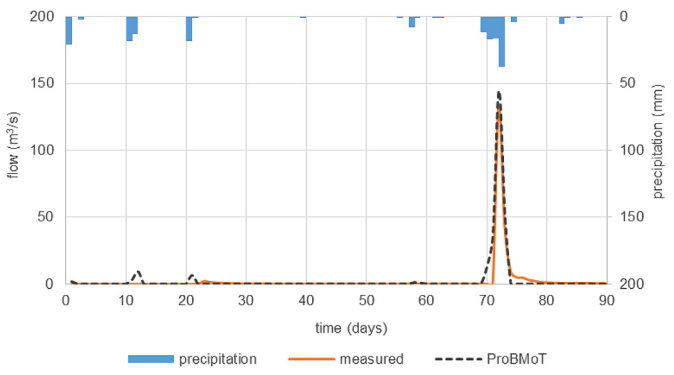


Fig. 13. Model validation results: a comparison between the measured and simulated outflows at the Tenencia hydrological station.

Table 1
Model evaluation statistics and performance (according to Moriasi et al., 2007) for both (calibration and validation) modeling phases.

	Calibration	Validation
R^2	0.99 (very good)	0.90 (very good)
NSE	0.99 (very good)	0.93 (very good)
PBIAS	14.29 (good)	−22.98 (satisfactory)
RSR	0.09 (very good)	0.32 (very good)

formalism, all the entities involved in a specific process are defined within the process specification. Therefore, the interplay of entities and processes is evident and clear. In Lagramge, a complete library has to be analyzed in detail in order to disclose such interactions.

In addition, ProBMoT provides means for logical grouping of entities and processes into compartments. Furthermore, it offers full simulation of differential equations supporting unobserved (hidden) variables. In watershed modeling, this feature can be crucial since having a completely observable system is usually an unrealistic scenario.

Although the main purpose of the proposed modeling system is building of alternative models and discovery of the best one, it can also be used for the other purposes. If the user is not interested in searching for different model structures and would rather like to test the advanced optimization methods incorporated into ProBMoT, he/she can limit the search space of mathematical models considered by ProBMoT by selecting the preferred mathematical formulation when specifying the conceptual model.

However, there are some disadvantages associated with the presented formalism. Firstly, it does not support the encoding of difference equations and conditional statements (e.g. if-then rules). Secondly, all the equations included in the library have to use the same (e.g. daily) time step. Consequently, it is difficult to integrate a large spectrum of legacy watershed models into the library. Another drawback is that all the models need to be consistently encoded in order to be utilized by the automated modeling tool. In order to take full advantage of the proposed AM methodology, more alternative formulations need to be encoded. And finally, the modeling system lacks a graphical user interface, which limits its use in a wider context. All of these issues will be addressed in future work.

The formalism we use is open for integrating knowledge from other domains, e.g., population dynamics and water quality modeling of aquatic ecosystems. This kind of knowledge already exists in the previously mentioned aquatic ecosystem library (Atanasova et al., 2006) and has been rewritten in the formalism proposed here (Čerepnalkoski et al., 2012). Thus, the two libraries could be integrated in two different ways. Because they are written in the same formalism and are using the same (daily) time step, they could be easily joined together in a single library. Such a library could be used for modeling water cycle and nutrient loading processes on a watershed scale, as well as for modeling biochemical processes in related aquatic ecosystems. This kind of approach could be considered as tight integration. The other possibility would be to use loose integration, by taking the results obtained with the watershed models (generated on the basis of the watershed modeling library), e.g., flow, nutrient loadings, etc., as input for the generation of aquatic ecosystem models.

We believe that the hierarchical organization of the watershed processes and their alternative formulations in one library can contribute towards more transparent and consistent encoding of watershed models, easier knowledge sharing among modelers, and easier (automatic or manual) model building for specific tasks.

6.2. Other efforts and their relation to ProBMoT

The AM methodology we present is related to the recent approaches of semantic environmental modeling (Villa et al., 2009). These approaches rely on the use of ontologies that define the basic concepts in the studied environmental domain and the relations between them. The ontologies are used to annotate environmental datasets, on one hand, and environmental software (models), on the other hand. Such annotations allow environmental models to be treated as services in the context of service-oriented architectures (Granell et al., 2010; Goodall et al., 2011). The main goal of the model as service concept is to make models and their outputs more accessible, to increase interoperability, and to work towards a larger vision of systems of interacting models (Nativi et al., 2013).

In our approach, the models are annotated by virtue of the ontological nature of the process-based modeling formalism. Not only are the executable models annotated, but so are their components (compartments, entities and processes), as well as libraries of domain knowledge and datasets. This allows them to be treated as resources in a service-oriented framework and to be discovered for reuse by queries specifying some of their properties.

Because models are often more useful when connected to other models, they are being increasingly integrated into larger simulation frameworks offering different capabilities. In such frameworks, the components (modules) exist as individual pieces of software that can be run in a stand-alone fashion. The composition of modules into more complex models proceeds manually through the use of interfaces.

When connecting a specific legacy model into a framework, it can happen that there is just a portion of the model that is of interest. This often requires decoupling of the selected model into more basic components (Holzworth et al., 2010). To allow for finer granularity of representation, our framework was designed based on such basic components (namely entities and processes involved in the domain of study) and not complete executable models. The structure of components is transparent and the representation formalism uniform across them. This allows for the automated construction of new models from the given components (or the reconstruction of existing legacy models) that fit the provided data.

Thus, a major difference between our framework and the other integrated environmental modeling frameworks is that our approach enables automatic generation of all possible model structures and automatic selection of the "optimal" model for the selected study area (based on calibration). To our knowledge, all the other approaches offer only manual selection of the alternative formulations (or alternative modules) for the selected processes. In this case, the user has to select the modules he/she wants to combine in order to implement the model and must be careful to use mutually consistent models that match the complexity of the modeled system.

7. Conclusions and further work

In this paper, we presented a new methodology for watershed modeling based on a knowledge library and a machine learning approach to automated modeling. The methodology establishes a new framework for testing alternative watershed model structures, where instead of selecting an existing model to be applied to a specific watershed or developing a new one, an automated search among different formulations of watershed processes (encoded in the knowledge library) is enabled. The search is guided by the modeler by specifying a conceptual model of the observed system at the desired level of detail.

The methodology was applied to the Foupana River catchment to extract a semi-distributed hydrological model. By specifying the conceptual model and using the knowledge library, two viable hydrological models were generated for the selected study area, differing in the formulations used for the calculation of PET. Further on, the two models were automatically calibrated against measurements. The model with the lower RMSE value was chosen as the best hydrological model (among the two alternative ones) for the selected study area.

Future work will focus on the application of the proposed AM methodology to modeling larger-scale watersheds. Several optimization methods within the AM framework will be applied to select the best hydrological and nutrient loading model structure

and the most suitable values of the selected parameters. Given the best model structure, various impact scenarios of altering water and land management practices on the quality of the related water bodies will be tested.

Furthermore, we will expand the library with biogeochemical transformation processes, as well as with more alternative formulations for the processes. We also foresee model building support beyond the modeling of water cycle and nutrient loadings by linking the library presented here to libraries covering other domains, thus enabling integrated modeling of watersheds and aquatic ecosystems. This would require the consideration of many new issues, such as the feedback between models and distributing data from a single source across the system.

We are also considering the possibility of treating the models that result from the automated model building procedure as web services in the context of service-oriented architectures. This would allow other researchers to access the generated models and to use them in other frameworks.

Acknowledgments

The first author is part-financed by the European Union, European Social Fund, under contract number P-MR-08/23. Džeroski is supported by the Slovenian Research Agency (Grant P2-0103), the European Commission (Grants ICT-2010-266722 and ICT-2011-287713), and Operation no. OP13.1.1.2.02.0005 financed by the European Regional Development Fund (85%) and the Slovenian Ministry of Higher Education, Science and Technology (15%).

The authors would like to thank the Administração de Região Hidrográfica do Algarve (ARH Algarve) for providing the boundaries for the selected experimental subcatchments, the Corine land cover data and the soil map. Special thanks go to Radhouane Ben Hamadou and Pedro Range (Centro de Ciências do Mar – CCMAR) for their assistance in data acquisition.

Appendix A. Supplementary data

Supplementary data related to this article can be found at http://dx.doi.org/10.1016/j.envsoft.2013.12.017.

References

Argent, R.M., 2004. An overview of model integration for environmental applications – components, frameworks and semantics. Environ. Model. Softw. 19 (3), 219–234.
Argent, R.M., Perraud, J.M., Rahman, J.M., Grayson, R.B., Podger, G.M., 2009. A new approach to water quality modeling and environmental decision support systems. Environ. Model. Softw. 24, 809–818.
Arnold, J.G., Fohrer, N., 2005. SWAT2000, current capabilities and research opportunities in applied watershed modeling. Hydrol. Process. 19 (3), 563–572.
Atanasova, N., Todorovski, L., Džeroski, S., Kompare, B., 2006. Constructing a library of domain knowledge for automated modeling of aquatic ecosystems. Ecol. Model. 194, 14–36.
Atanasova, N., Todorovski, L., Džeroski, S., Kompare, B., 2008. Application of automated model discovery from data and expert knowledge to a real-world domain, Lake Glumsø. Ecol. Model. 212, 92–98.
Atanasova, N., Džeroski, S., Kompare, B., Todorovski, L., Gal, G., 2011. Automated discovery of a model for dinoflagellate dynamics. Environ. Model. Softw. 26 (5), 658–668.
Behrendt, H., Huber, P., Kornmilch, M., Opitz, D., Schmoll, O., Scholz, G., Uebe, R., 1999. Nutrient Emissions into River Basins of Germany. UFOPlan-Ref. No. 296 25 515. UBA, Berlin.
Čerepnalkoski, D., Taskova, K., Todorovski, L., Atanasova, N., Džeroski, S., 2012. The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. Ecol. Model. 245, 136–165.
Donigian, A.S., Bicknell, B.R., Imhoff, J.C., 1995. Hydrological simulation program – Fortran (HSPF). In: Singh, V.P. (Ed.), Computer Models of Watershed Hydrology. WRP, Highlands Ranch, CO, pp. 395–442.
David, O., Ascough II, J.C., Lloyd, W., Green, T.R., Rojas, K.W., Leavesley, G.H., Ahuja, L.R., 2013. A software engineering perspective on environmental modeling framework design: the Object Modeling System. Environ. Model. Softw. 39, 201–213.
Driver, N.E., Tasker, G.D., 1988. Techniques for Estimation of Storm-runoff Loads, Volumes, and Selected Constituent Concentrations in Urban Watersheds in the United States. USGS Open-File Report 88-191, Denver, CO.
Džeroski, S., Todorovski, L., 2003. Learning population dynamics models from data and domain knowledge. Ecol. Model. 170, 129–140.
Granell, C., Díaz, L., Gould, M., 2010. Service-oriented applications for environmental models: reusable geospatial services. Environ. Model. Softw. 25 (2), 182–198.
Gregersen, J.B., Gijsbers, P.J.A., Westen, S.J.P., 2007. OpenMI, open modeling interface. J. Hydroinform. 9, 175–191.
Goodall, J.L., Robinson, B.F., Castronova, A.M., 2011. Modeling water resource systems using a service-oriented computing paradigm. Environ. Model. Softw. 26 (5), 573–582.
Gupta, H.V., Sorooshian, S., Yapo, P.O., 1999. Status of automatic calibration for hydrologic models: comparison with multilevel expert calibration. J. Hydrol. Eng. 4 (2), 135–143.
Haith, D.A., 1985. An event-based procedure for estimating monthly sediment yields. Trans. ASAE 28 (6), 1916–1920.
Haith, D.A., Shoemaker, L.L., 1987. Generalized watershed loading functions for stream flow nutrients. Water Resour. Bull. 23, 471–478.
Haith, D.A., Mandel, R., Shyan Wu, R., 1992. GWLF – Generalized Watershed Loading Functions, Version 2.0. User's Manual. Cornell University, Ithaca, NY.
Hamon, W.R., 1961. Estimating potential evapotranspiration. Proc. Am. Soc. Civ. Eng. J. Hydraul. Div. 87 (HY3), 107–120.
Hargreaves, G.L., Hargreaves, G.H., Riley, J.P., 1985. Agricultural benefits for Senegal River basin. J. Irrig. Drain. Eng. 111 (2), 113–124.
Holzworth, D.P., Huth, N.I., de Voil, P.G., 2010. Simplifying environmental model reuse. Environ. Model. Softw. 25, 269–275.
Laniak, G.F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M., Peckam, S., Reaney, S., Gaber, N., Kennedey, R., Hughes, A., 2013. Integrated environmental modeling: a vision and roadmap for the future. Environ. Model. Softw. 39, 3–23.
Leavesley, G.H., Markstrom, S.L., Restrepo, P.J., Viger, R.J., 2002. A modular approach to addressing model design, scale, and parameter estimation issues in distributed hydrological modeling. Hydrol. Process. 16, 173–187.
Moriasi, D.N., Arnold, J.G., Van Liew, M.W., Bingner, R.L., Harmel, R.D., Veith, T.L., 2007. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. Trans. ASABE 50 (3), 885–900.
Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models part I – a discussion of principles. J. Hydrol. 10 (3), 282–290.
Nativi, S., Mazzetti, P., Geller, G.N., 2013. Environmental model access and interoperability: the GEO Model Web initiative. Environ. Model. Softw. 39, 214–228.
Schwarz, G.E., Hoos, A.B., Alexander, R.B., Smith, R.A., 2006. The SPARROW surface water-quality model, theory, application and user documentation. In: U.S. Geological Survey Techniques and Methods. USGS, Reston, VA. Book 6, Section B, (Chapter 3).
Soil Conservation Service Engineering Division, 1986. Urban Hydrology for Small Watersheds. U.S. Department of Agriculture. Technical Release 55.
Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. 11, 341–359.
Todorovski, L., Bridewell, W., Shiran, O., Langley, P., 2005. Inducing hierarchical process models in dynamic domains. In: Proceedings of the Twentieth National Conference on Artificial Intelligence. AAAI Press, Menlo Park, CA, pp. 892–897.
US EPA, 1999. Protocols for Developing Nutrient TMDLs. EPA 841- B-99–007. Office of Water (4503 F), Washington, DC.
Villa, F., Athanasiadis, I.N., Rizzoli, A.E., 2009. Modelling with knowledge: a review of emerging semantic approaches to environmental modeling. Environ. Model. Softw. 24, 577–587.
Voinov, A., Costanza, R., Wainger, L.A., Boumans, R.M.J., Villa, F., Maxwell, T., Voinov, H., 1999. Patuxent landscape model, integrated ecological economic modeling of a watershed. Environ. Model. Softw. 14, 473–491.
Voinov, A., Fitz, C., Boumans, R., Costanza, R., 2004. Modular ecosystem modeling. Environ. Model. Softw. 19, 285–304.
Welsh, W.D., Vaze, J., Dutta, D., Rassam, D., Rahman, J.M., Jolly, I.D., Wallbrink, P., Podger, G.M., Bethune, M., Hardy, M.J., Teng, J., Lerat, J., 2013. An integrated modeling framework for regulated river systems. Environ. Model. Softw. 39, 81–102.
Williams, J.R., 1995. The EPIC model. In: Singh, V.P. (Ed.), Computer Models of Watershed Hydrology. Water Resources Publications, Highlands Ranch, CO, pp. 909–1000.
Wischmeier, W.H., Smith, D.D., 1978. Predicting Rainfall Erosion Losses – a Guide to Conservation Planning. In: Agricultural Handbook 537. U.S. Department of Agriculture, Washington, DC.
Wu, C., Cheng, C., Lo, H., Chen, Y., 2010. Study on estimating the evapotranspiration cover coefficient for stream flow simulation through remote sensing techniques. Int. J. Appl. Earth Obs. Geoinf. 12, 225–232.