A knowledge-based method for the automatic determination of hydrological model structures

Jingchao Jiang, A-Xing Zhu, Cheng-Zhi Qin and Junzhi Liu

ABSTRACT

To determine a suitable hydrological model structure for a specific application context using integrated modelling frameworks, modellers usually need to manually select the required hydrological processes, identify the appropriate algorithm for each process, and couple the algorithms' software components. However, these modelling steps are difficult and require corresponding knowledge. It is not easy for modellers to master all of the required knowledge. To alleviate this problem, a knowledge-based method is proposed to automatically determine hydrological model structures. First, modelling knowledge for process selection, algorithm identification, and component coupling is formalized in the formats of the Rule Markup Language (RuleML) and Resource Description Framework (RDF). Second, the formalized knowledge is applied to an inference engine to determine model structures. The method is applied to three hypothetical experiments and a real experiment. These experiments show how the knowledge-based method could support modellers in determining suitable model structures. The proposed method has the potential to reduce the knowledge burden on modellers and would be conducive to the promotion of integrated modelling frameworks.

Key words | hydrological modelling, integrated modelling frameworks, knowledge-based method, modelling knowledge

Jingchao Jiang
Smart City Research Center, School of Automation,
Hangzhou Dianzi University,
Hangzhou 310012,
China

A-Xing Zhu
Department of Geography,
University of Wisconsin–Madison,
Madison, WI 53706,
USA

Cheng-Zhi Qin (corresponding author)
State Key Laboratory of Resources and Environmental Information System,
Institute of Geographic Sciences and Natural Resources Research, CAS,
Beijing 100101,
China
E-mail: *qincz@lreis.ac.cn*

Junzhi Liu
Key Laboratory of Virtual Geographic Environment, Ministry of Education,
Nanjing Normal University,
Nanjing 210023,
China

INTRODUCTION

It is widely recognized that there is no universal hydrological model that can be applied in all application contexts (WMO 1975, 1992). To better understand the natural and human influences on watersheds, users usually need to determine a model structure that is suitable for the specific application context (Rigon *et al.* 2006; Baymani-Nezhad & Han 2013; Furusho *et al.* 2013; Lai *et al.* 2016; Vo & Gourbesville 2016). Integrated modelling frameworks are designed to support model construction through the dynamic coupling of fine-grained components. Examples of such frameworks include the Open Modelling Interface (OpenMI) (Moore & Tindall 2005), the Community Modelling Systems (CMS) (Lu & Piasecki 2012), the Object Modelling System (OMS) (David *et al.* 2013), the Community Surface Dynamics Modelling System (CSDMS) (Peckham *et al.* 2013), the Spatially Explicit Integrated Modelling System (SEIMS) (Liu *et al.* 2016), and the HydroInformatic Modelling System (HIMS) (Wang *et al.* 2018).

To determine a suitable model structure, modellers usually need to manually select the required processes, identify the appropriate algorithm for each process, and couple the software components of the algorithms. However, these modelling steps are difficult and always require corresponding knowledge (Elag & Goodall 2013; Peckham *et al.* 2013). It is not easy for modellers to master all of the required modelling knowledge.

doi: 10.2166/hydro.2019.029

Knowledge-based methods can be used to reduce the knowledge burden for hydrological modelling (Liu *et al.* 2013; Ward *et al.* 2019). Recently, knowledge-based methods, which incorporate modelling knowledge into integrated modelling frameworks, have been proposed to make component coupling or model selection easier.

To make component coupling easier, Islam & Piasecki (2008) proposed an ontology for the metadata of numerical models based on the Web Ontology Language (OWL), which can support the coupling of different components. Elag & Goodall (2013) presented an ontology for describing the core concepts and relationships for hydrological modelling, which could be further used for component selection and coupling. For automatic coupling of alternative algorithms for watershed modelling, Škerjanec *et al.* (2014) developed a knowledge library that described hydrological processes, calculation formulas, and their input/output variables using a domain-specific language. Peckham (2014) proposed a smart modelling framework for component coupling by a standardized model interface and corresponding metadata. Harpham & Danovaro (2015) sought to design standard metadata to describe environmental numerical models and their interfaces to other models. The standard metadata could be used for model coupling. Morsy *et al.* (2017) designed a model metadata framework, in which metadata elements were expressed as Resource Description Framework (RDF) triples, to support the sharing and reuse of hydrological models. Jiang *et al.* (2017) developed a service-oriented modelling framework to couple models based on the Basic Model Interface (BMI) (Peckham *et al.* 2013).

To make model selection easier, Chau (2007) proposed an ontology-based knowledge management system to assist users with the selection of the appropriate models for flow and water quality modelling. Qiu *et al.* (2017) proposed an ontology-based approach to describe environmental models and disaster-related data through semantics. Based on the ontology-based approach, the flood management system could recommend suitable models for users to apply when constructing a workflow.

The previous work mainly emphasized the usage of component-coupling knowledge or model-selection knowledge, whereas little attention has been paid to the usage of process-selection and algorithm-identification knowledge. It is still difficult for modellers to select appropriate processes and algorithms. Moreover, if modellers do not master the relevant knowledge, unsuitable models might be built (Voinov & Shugart 2013).

To alleviate this problem, this paper proposes a knowledge-based method to automatically determine hydrological model structures. Note that this study focuses on the determination of hydrological model structures. Although data preprocessing and parameter calculation are necessary after the hydrological model structure is determined, these topics are outside the scope of this study.

METHODS

Design of the knowledge-based method

To determine hydrological model structures automatically, a knowledge-based method is proposed (Figure 1). The framework of this method consists of three steps. First, process-selection and algorithm-identification knowledge are obtained

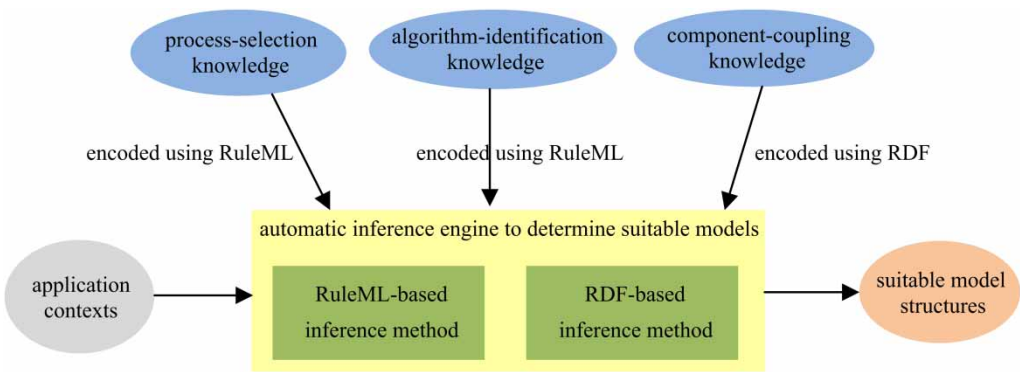

Figure 1 | Framework of the knowledge-based method for the automatic determination of hydrological model structures.

from experts and literature, while component-coupling knowledge is extracted from the metadata of specific modelling frameworks. Second, the obtained knowledge is formally encoded using the Rule Markup Language (RuleML) (Boley *et al.* 2010) and RDF (Cyganiak *et al.* 2014), which can be used by an inference engine. Third, an inference engine is designed and implemented according to the typical procedures of model structure determination, and it is used to generate model structures. Each of the three steps will be discussed in detail in the following subsections.

Preparation of hydrological modelling knowledge

Knowledge of hydrological process selection

A hydrological model consists of multiple hydrological processes such as infiltration/surface runoff, depression, subsurface flow, snowmelt, and groundwater processes. Whether a hydrological process should be involved depends on the simulation purpose (i.e. expected model output), climatic and underlying surface conditions, geological conditions, and hydrological conditions of the region, as well as spatial and temporal scales. For example, when simulating a short periodic rainfall-runoff event in an arid region, where the aeration zone is not easily saturated, subsurface flow and groundwater processes can be omitted. To simulate the water balances in alpine regions where snowmelt might be a source of the discharge peak and a major cause of flooding, the snowmelt process should be involved. If special hydrological or geological conditions such as frozen soil or glaciers exist in a watershed, corresponding processes should be considered. Information regarding which processes are needed for certain application contexts can be listed. This type of knowledge can be obtained from hydrologists and literature.

Knowledge of algorithm identification

There are multiple algorithms that can be used to simulate one hydrological process, and each algorithm has specific application conditions. Algorithm selection depends on watershed physiographic conditions, temporal scale, and data availability.

Watershed physiographic conditions. Every algorithm has assumptions and can be used only under specific physiographic conditions. For example, regarding the infiltration/surface runoff process, the infiltration excess algorithm should be selected for arid watersheds, whereas the saturation excess algorithm should be selected for humid watersheds.

Temporal scale. The temporal scale should be considered in time-based physical algorithms. For example, the Soil and Water Assessment Tool (SWAT) contains two methods for simulating the surface runoff process, i.e. the Soil Conservation Service Curve Number (SCS-CN) method and the Green–Ampt method. The SCS-CN method is suitable for simulations at a daily scale, while the Green–Ampt method is suitable for simulations at an hourly scale or finer time steps (Grimaldi *et al.* 2013).

Data availability. Data availability limits the applicability of an algorithm. In real applications, due to the lack of data, it is common to replace an algorithm that can simulate a process accurately using detailed data with an algorithm that has fewer input data requirements. For example, among the methods for simulating the potential evapotranspiration (PET) process, the Penman–Monteith method requires the mean daily temperature, relative humidity, solar radiation, and wind speed as input (Allen *et al.* 1998); the method proposed by Hargreaves and Samani requires daily or monthly maximum and minimum temperatures as input (Hargreaves & Samani 1985), while Thornthwaite's method takes only the mean monthly temperature as input (Thornthwaite 1948). If only data for the mean monthly temperature are available, Thornthwaite's method can be applied.

Knowledge of component coupling

One hydrological modelling component is the software component of a hydrological algorithm in a specific modelling framework. There may exist slight differences in the input/output interfaces for different components of the same algorithm. After identifying the appropriate algorithm for each process, it is necessary to check whether the components of these algorithms are compatible and could be used to assemble a complete composite model (i.e. whether every component can obtain its input from the existing input data or other components' output) (Peckham 2014). For each component, the corresponding algorithm and its input and output should be clearly stated.

Encoding hydrological modelling knowledge

For the unified description and reuse of modelling knowledge, a standard-name library was built (accessible on https://github.com/lreis2415/SEIMS/tree/master/knowledge/rdfBase/variable). The library includes the names of processes, algorithms, variables, and keywords of application context descriptions. These standard names are used to encode hydrological modelling knowledge. Process-selection and algorithm-identification knowledge can be naturally expressed as conditional sentences in the form of 'if …, then …'. Knowledge of this type is called procedural knowledge or a production rule (Anderson 1983). Component-coupling knowledge can be expressed as a statement called declarative knowledge (Anderson 1983). The encoding form for each type of hydrological modelling knowledge is described below.

Encoding process-selection knowledge and algorithm-identification knowledge

Both process-selection knowledge and algorithm-identification knowledge are types of procedural knowledge, and they can be encoded using RuleML, which is a markup language designed for the interchange of web rules in an XML format. The language is uniform across various rule languages and platforms (Boley *et al.* 2010). Figures 2 and 3

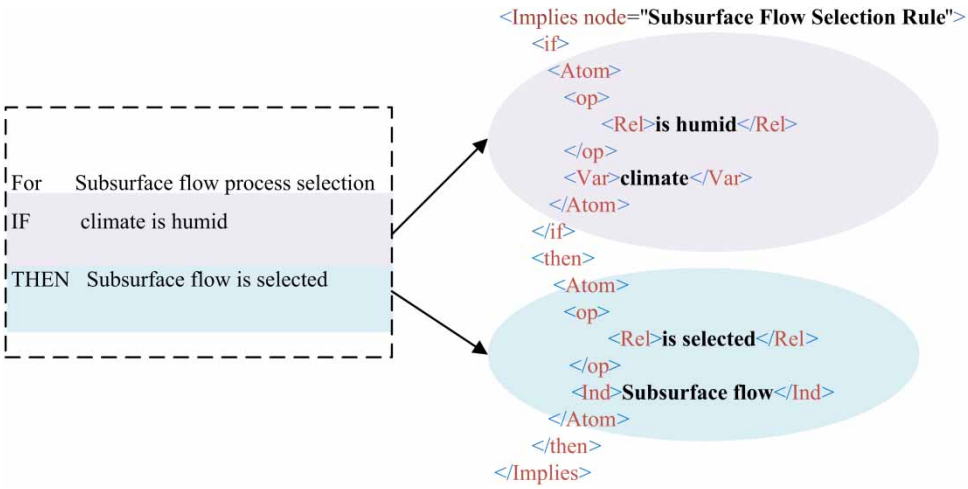

Figure 2 | Example of encoding the subsurface flow process-selection knowledge in RuleML.

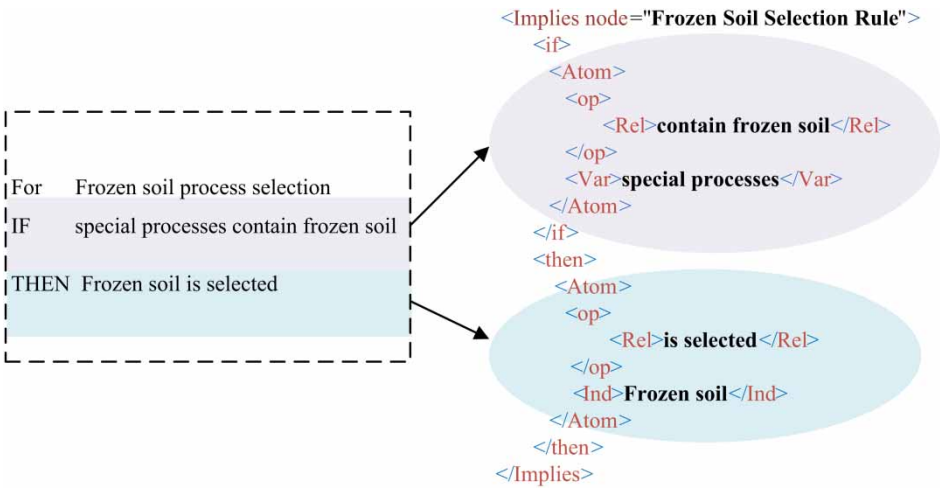

Figure 3 | Example of encoding the frozen soil process-selection knowledge in RuleML.

show two examples of encoding process-selection knowledge in RuleML. Figure 4 provides an example of encoding the algorithm-identification knowledge for the surface runoff process. For a more detailed description of RuleML, please refer to Boley *et al.* (2010).

Encoding component-coupling knowledge

Hydrological component-coupling knowledge is declarative; thus, the knowledge can be treated as a statement and formalized as an object-attribute-value triple. The triple can be encoded using RDF, a W3C standard for describing identifiable resources (Cyganiak *et al.* 2014). For example, the component 'SCS-CN_Com' has NEPR (i.e. net precipitation), DEPST (i.e. depression storage), and SOTE (i.e. soil temperature) as input variable names, it has EXCP (i.e. excess precipitation), SOMO (i.e. average soil moisture), and INFIL (i.e. infiltration) as output variable names, and its algorithm name is SCS-CN. Figure 5 gives an example of encoding the component 'SCS-CN_Com' in RDF. For a more detailed description of RDF, please refer to Cyganiak *et al.* (2014).

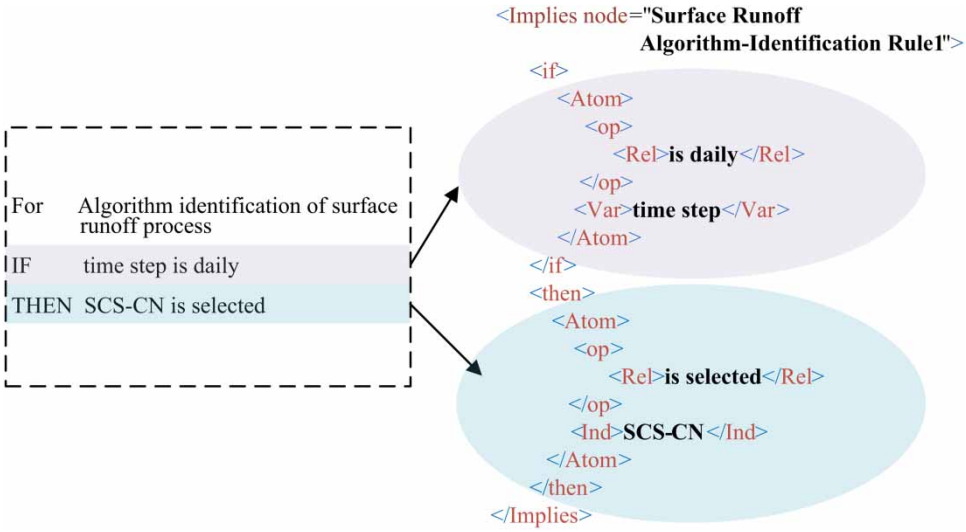

Figure 4 | Example of encoding the algorithm-identification knowledge for the surface runoff process in RuleML.

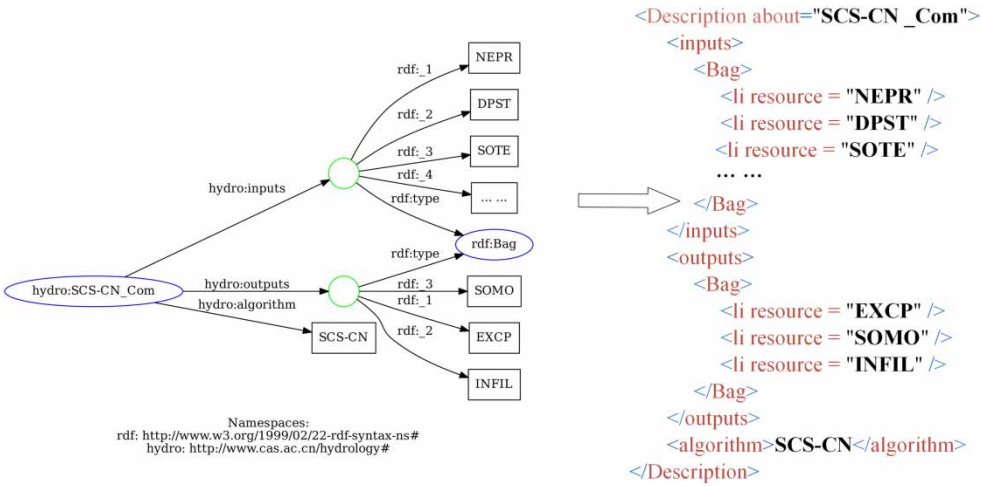

Figure 5 | Example of encoding the component-coupling knowledge for a component of the SCS-CN algorithm in RDF.

Inference engine

There are three main steps for the inference engine to determine hydrological model structures (Figure 6).

The first step is to select m required hydrological processes through the RuleML-based inference method according to the application purpose, spatial and temporal scales, and watershed physiographic conditions. The collection of selected processes can be considered to be an abstract conceptual model. The second step is to identify n_i $(i = 1, 2, \ldots, m;\ n_i \geq 1)$ appropriate algorithms for each process through the RuleML-based inference method according to the watershed physiographic conditions, data availability, and time step. The collections of these selected

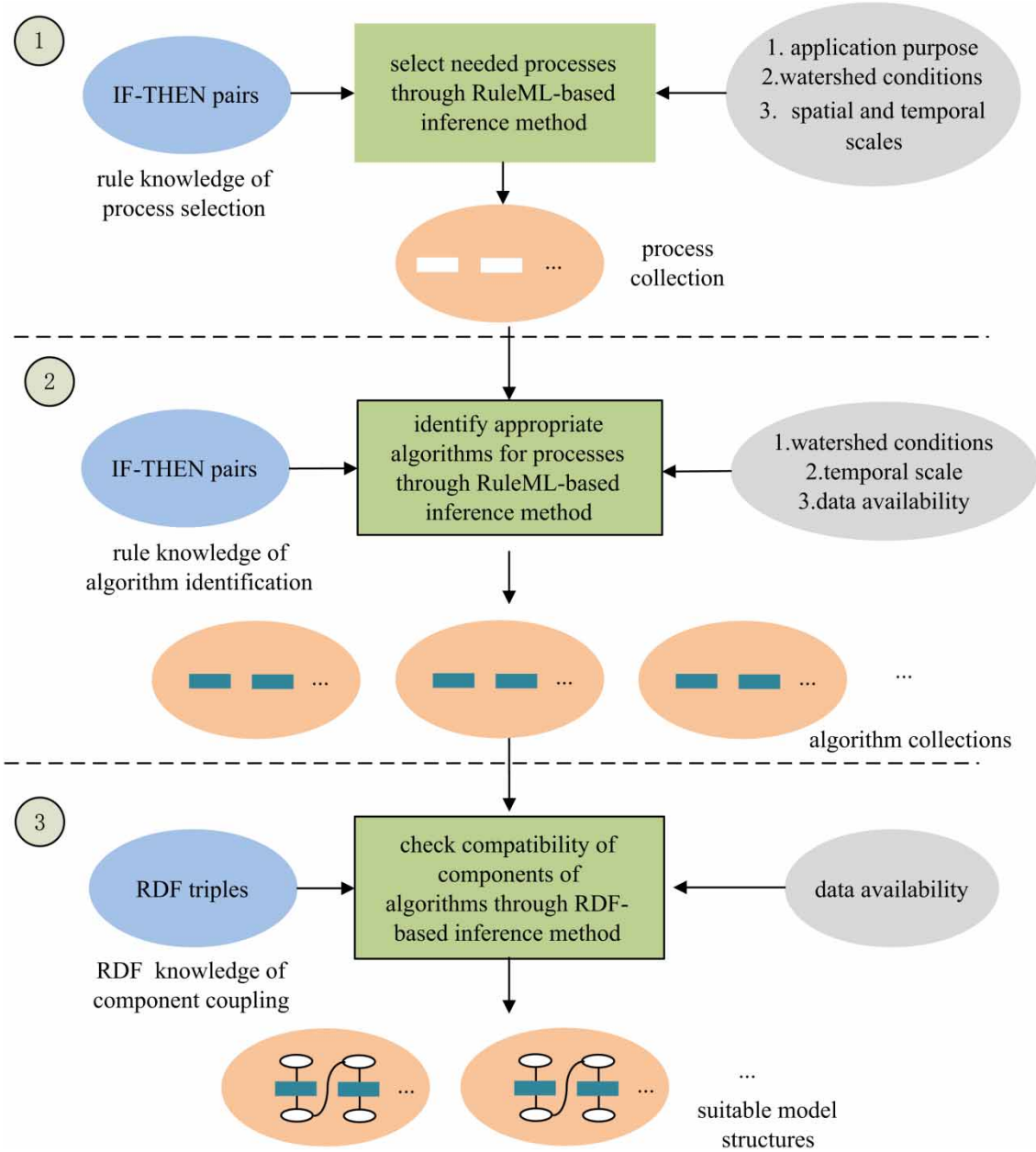

Figure 6 | Procedure for the inference engine to determine hydrological model structures.

algorithms can be considered to be specific conceptual models. This study assumes that each algorithm corresponds to only one component in the specific modelling framework. Therefore, there are a total of $\prod_{i=1}^{m} n_i$ component combinations. The third step is to check whether all the inputs for every component in a component combination are available. If so, the component combination will be selected as an alternative one. If not, the component combination cannot be executed and will be excluded. If there are multiple alternative combinations, the inference engine will select one of them as the optimal one. Then, the components of this optimal component combination are sequenced into an optimal workflow. The source code of the inference engine is available on GitHub (https://github.com/lreis2415/SEIMS/tree/master/knowledge/inference_code). The detailed designs of the RuleML-based and RDF-based inference methods are as follows.

RuleML-based inference method for process selection and algorithm identification

The RuleML-based inference method is designed to be forward chaining. Taking process selection as an example to illustrate the steps of the RuleML-based inference method, the first step is to load the process-selection knowledge from the knowledge base and store each rule in a 'ruleml' class object. The fields of the 'ruleml' class include the name of the rule, the 'if' part and the 'then' part. The 'if' and 'then' parts are object instances of the 'atom' class. The fields of the 'atom' class include 'var' (variables), 'rel' (predicates), and 'ind' (constants). The second step is to initialize the application context and to store it in a HashMap <context keyword, context value> *map_c*. The third step is to determine for each 'ruleml' object whether the 'if' part can be triggered by the application context. This step aims to determine whether the 'var' value obtained by *map_c* can match the 'rel' value of the 'if' part. If the 'if' part is triggered, then the 'then' part is used to determine whether the corresponding hydrological process is selected. That is, according to the semantic description of 'rel', it is determined whether the hydrological process described by the 'ind' in 'then' parts is selected. For example, by using the RuleML-based inference method to the process-selection knowledge (Figure 2), the subsurface flow process will be selected if the climate type value in the application contexts is 'humid'.

RDF-based inference method for component coupling

The RDF-based inference method is used to determine whether the components of the selected algorithms are compatible and can be used to assemble a complete composite model. The RDF-based inference method consists of three steps. The first step is to load the RDF knowledge of all components from the knowledge base and store the RDF knowledge in a HashMap <algorithm name, 'rdf' class object> *map_a*. The fields of the 'rdf' class include component name, algorithm name, input variable names, and output variable names. The second step is to obtain the 'rdf' object set according to *map_a* for a given algorithm name set. The third step is to determine whether each input variable of the object can be satisfied (i.e. whether the data required for each input variable can be obtained from the existing data or the output of other components) for each 'rdf' object. Specifically, for each input variable, if the name of the input variable is included in the existing data names or the output variable names of other 'rdf' objects, the input variable can be satisfied; otherwise, the input variable cannot be satisfied. If all the input variables of each 'rdf' object can be satisfied, the corresponding component set can be coupled into a workflow; otherwise, the corresponding component set will be excluded.

The components are sorted by the dataflow among components. The dataflow can be abstracted as a directed graph. For a dataflow without loops, a directed acyclic graph (DAG) is used to describe the dataflow. Each component is treated as a vertex of the DAG, and the input–output relationship of the variables between two components is treated as a unidirectional edge of the DAG. The topological sorting method is used to determine the linear ordering of the DAG's vertices (i.e. the execution sequence of the components). For a dataflow with loops, which can be described by a directed cyclic graph (DCG), each loop needs to be broken to convert the DCG into a DAG.

Figure 7 shows an example of the cyclic dependency in coupling SUR_EXCESS (i.e. runoff yield under the excess infiltration component) with DEP_FS (i.e. depression component). SUR_EXCESS needs DPST (i.e. distribution of

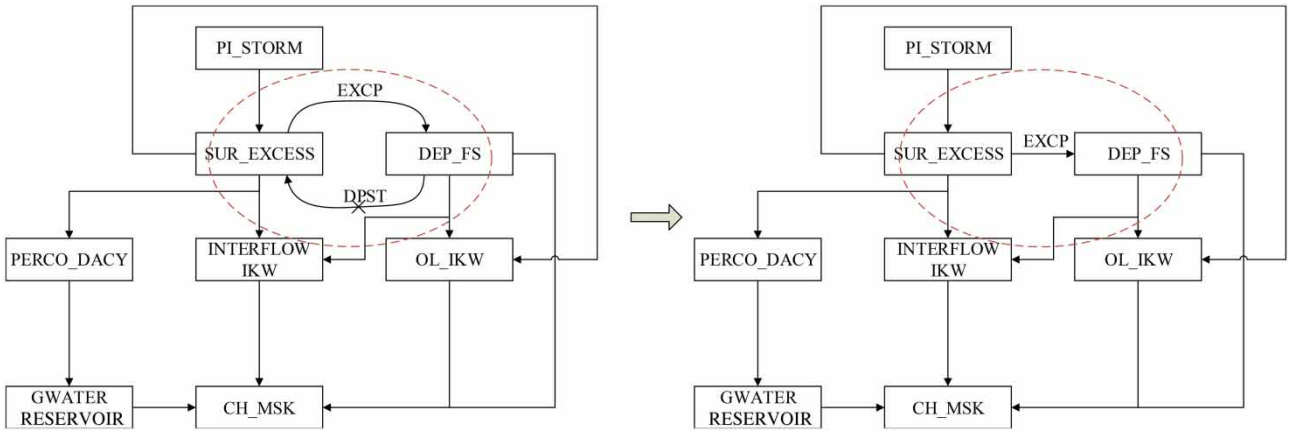

Figure 7 | Example of converting DCG into DAG.

depression storage) as an input, which is the output of the DEP_FS. Meanwhile, DEP_FS needs EXCP (i.e. excess precipitation) as an input, which is the output of SUR_EXCESS. Due to this loop between SUR_EXCESS and DEP_FS, the sequence between these two components is undecidable. Note that hydrologists usually assume that SUR_EXCESS happens before DEP_FS. Therefore, the loop between these two components can be broken by removing the 'DPST' edge from the current temporal step of the simulation. Thus, the DCG can be converted into a DAG, whose sequence can be determined by the topological sorting method. According to the DAG, the SUR_EXCESS and DEP_FS components will be executed sequentially within the current temporal step of the simulation. The output of DEP_FS, DPST, will be the input to SUR_EXCESS in the next temporal step of the simulation.

Applying the method to a specific modelling framework

In the proposed method, knowledge of process selection and algorithm identification is framework-independent (or generic knowledge), while metadata of components is framework-specific (or local knowledge). The metadata descriptions of the components in a specific modelling framework could be different from the ontological descriptions of the generic knowledge. To apply the proposed method to a specific modelling framework, the metadata descriptions of the local knowledge should be mapped to the ontological description of the generic knowledge. Specifically, the metadata information of each component, including the algorithm to which this component belongs and its input/output variables, is first extracted from the modelling framework. Second, the variable names and the algorithm names of the components are described using standard names, if they are inconsistent with the ones in the standard-names library. Then, the metadata of components is formalized as component-coupling knowledge and stored in the RDF knowledge base. Thus, the proposed method can be applied to a specific modelling framework.

RESULTS

Integrated modelling framework and modelling knowledge

To evaluate the applicability of the proposed knowledge-based method, we applied the method in a hydrological integrated modelling framework (i.e. SEIMS). SEIMS consists of a parallel hydrological module library, a runtime environment (including model initialization, model execution, and model calibration), auxiliary functions (such as preprocessing, postprocessing, and data management), and a hydrological database. The hydrological module library of SEIMS contains over 30 modules for the main hydrological processes covering hydrological, crop growth, and nutrient migration/transformation processes (Qin *et al.* 2018). Each module includes one or several components. These components can be coupled to build models. The source code

of SEIMS is available on GitHub (https://github.com/lreis2415/SEIMS).

Currently, there are a total of 47 rules and 208 RDF triples in the knowledge base (available at https://github.com/lreis2415/SEIMS/tree/master/knowledge).

Software prototype of the knowledge-based method

A software prototype called the Intelligent Hydrological Modelling Customizing System (IHMCS) has been developed as the shell of the knowledge-based method to assist modellers in the generation of hydrological model structures. IHMCS is a browser-based system (Figure 8), and it can be accessed on the URL (http://114.215.153.178:8080/IHMCS/main.jsp). This interface consists of a list of radio boxes and check boxes. These boxes represent the application context and are linked to the knowledge base. After users fill in the application context through the interface, the inference engine will be invoked to generate suitable model structures. The names of the selected hydrological processes, algorithms, and components are displayed in a grid form. The model structure can be saved in a configuration file in a format that SEIMS can utilize. It should be noted that the browser-side user interface is loosely coupled with the server-side inference engine. Different user interfaces can be designed to meet the needs of different users.

Experiments

Three hypothetical experiments and a real experiment are used to illustrate the capability of the knowledge-based method in the automatic determination of hydrological model structures.

Hypothetical experiments

Three different hypothetical experiments are designed. Each hypothetical experiment has an application context consisting of the application purpose, spatial and temporal scales, watershed conditions, data availability, and particular processes (Table 1).

As an example, the modelling context and model structure for the first hypothetical experiment are shown in Figure 8. The model structures for the three hypothetical experiments are listed in Tables 2–4, respectively.

During this model building procedure by IHMCS, the complicated professional details (i.e. selecting the processes, identifying the algorithms, and checking the compatibilities of the components) are transparent to the modellers. The modellers can easily obtain model structures. In addition, the model structures determined by IHMCS show that the proposed method can generate different model structures for different modelling contexts.

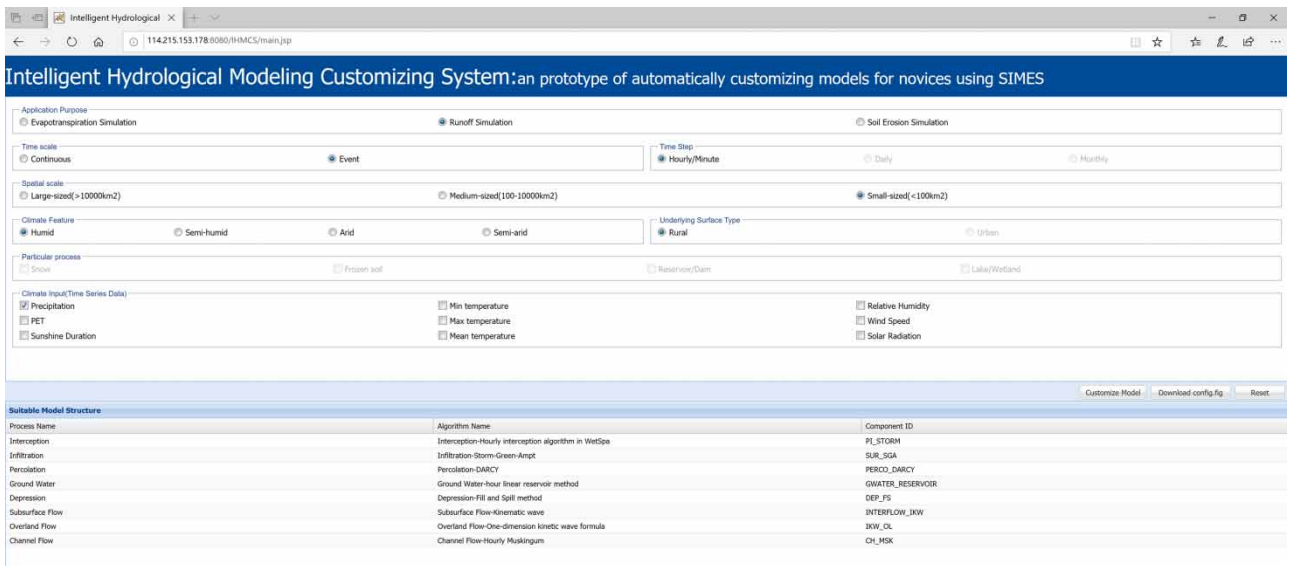

Figure 8 | Interface of IHMCS, and the application context and model structure of the first hypothetical experiment.

Table 1 | Hypothetical experiments

Experiment	Purpose	Time scale/step	Spatial scale	Watershed conditions	Data availability	Particular processes
Experiment 1	Rainfall-runoff	Event/hourly	Small-sized	Humid, rural region	Precipitation	
Experiment 2	Evapotranspiration	Continuous/daily	Medium-sized	Humid, rural region	Precipitation, minimum temperature, maximum temperature	Snow
Experiment 3	Soil erosion	Event/hourly	Small-sized	Semi-humid, rural region	Precipitation	

Table 2 | List of hydrological processes, algorithms, and components in the first hypothetical experiment

Process	Algorithm	Component ID
Interception	Hourly interception algorithm in WetSpa	PI_STORM
Infiltration	Green–Ampt method	SUR_SGA
Percolation	Darcy method	PERCO_DARCY
Ground water	Linear reservoir method	GWATER_RESERVOIR
Depression	Fill and Spill method	DEP_FS
Subsurface flow	Kinematic wave method	INTERFLOW_IKW
Overland flow	One-dimension kinetic wave formula	IKW_OL
Channel flow	Hourly Muskingum method	CH_MSK

Table 3 | List of hydrological processes, algorithms, and components in the second hypothetical experiment

Process	Algorithm	Component ID
PET	Hargreaves method	PET_H
Interception	Daily interception algorithm in WetSpa	PI_MSM
Snow melt	Degree-day method	SNO_DD
Soil temperature	Finn and Plauborg method	STP_FP
Infiltration	SCS-CN method	SUR_CN
Percolation	Brooks and Corey formula	PER_PI
Subsurface flow	Darcy formula	SSR_DA
Depression	Linsley method	DEP_LINSLEY
Soil evaporation	Evaporation method in WetSpa	SET_LM

Table 4 | List of hydrological processes, algorithms, and components in the third hypothetical experiment

Process	Algorithm	Component ID
Interception	Hourly interception algorithm in WetSpa	PI_STORM
Infiltration	Green–Ampt method	SUR_SGA
Percolation	Darcy method	PERCO_DARCY
Ground water	Linear reservoir method	GWATER_RESERVOIR
Depression	Fill and Spill method	DEP_FS
Subsurface flow	Kinematic wave method	INTERFLOW_IKW
Overland flow	One-dimension kinetic wave formula	IKW_OL
Splash erosion	Park equation	SplashEro_Park
Overland erosion	Erosion–Govers method	KinWavSed_OL
Channel flow	Hourly Muskingum method	CH_MSK
Channel erosion	Srinivasan Galvao function	KinWavSed_CH

Real experiment

Application context. This real experiment aims to simulate the daily rainfall-runoff in the Meichuan River watershed in Jiangxi Province, China (Figure 9). The watershed has a drainage area of approximately 6,366 km^2, with an elevation ranging from 151 to 1,425 m. The climate is humid and sub-tropical, with an average annual precipitation of 1,706 mm and a mean annual temperature of 17 °C. The lowest air temperature in the watershed is below 0 °C. The available spatial dataset includes a gridded digital elevation map, land-use map, and soil-type map. The dataset has a spatial

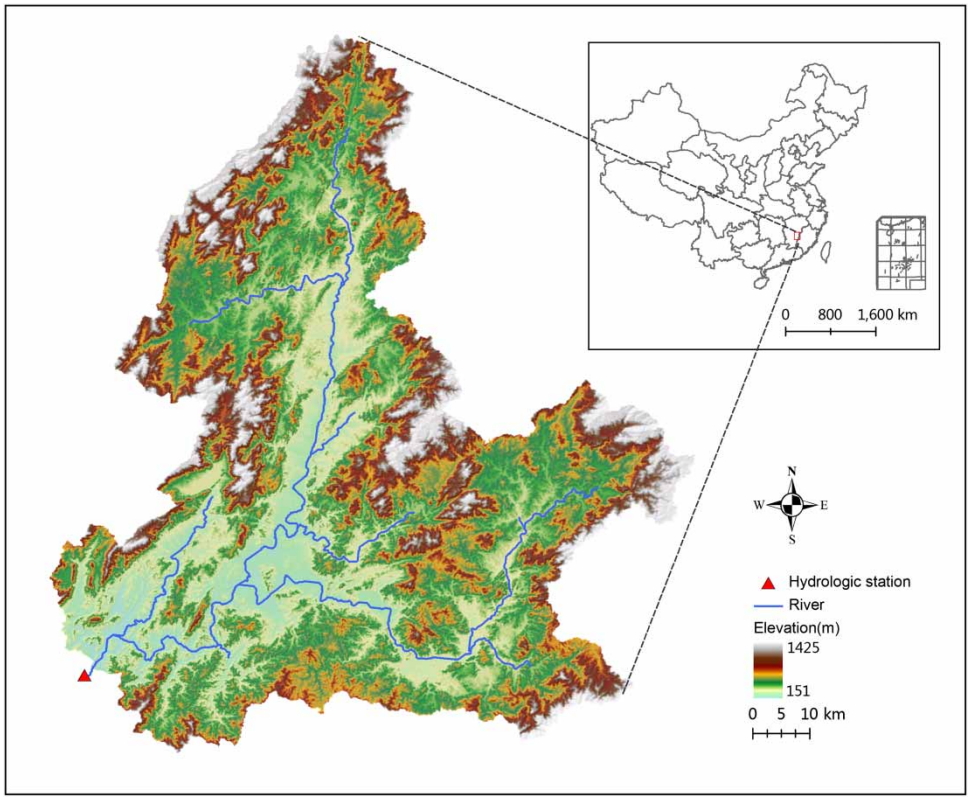

Figure 9 | Map of the Meichuan River watershed.

resolution of 90 m. The available time series dataset includes the precipitation, minimum temperature, maximum temperature, and discharge data at a daily time step from 1 January 2002 to 31 December 2005 and from 1 January 2007 to 31 December 2010. There was a lack of observation data in 2006.

Model structure. The modelling context and model structure for the real application context are shown in Figure 10. The processes, algorithms, and components of the model structure are listed in Table 5.

The related knowledge and application contexts for hydrological process selection and algorithm identification are shown in Tables 6 and 7, respectively. The RDF-based reasoning method confirms that the components of these selected algorithms can be assembled as a complete workflow.

Simulation results. The model parameters are prepared using the preprocessing tools in SEIMS. The model is calibrated using the discharge data from 2002 to 2005 and validated using the data from 2007 to 2010. Overall, it is found that the estimations agreed well with the observations, although the simulations in some certain high water periods are a little imprecise. The Nash–Sutcliffe efficiency (NSE) is used to evaluate the simulation accuracy. The NSE of the resulting hydrological model for the calibration period is 0.90 (Figure 11), and the NSE of the validation period is 0.88 (Figure 12). The simulation accuracy is acceptable.

DISCUSSION

The above results show how the proposed knowledge-based method could support modellers in the automatic determination of hydrological model structures for different application contexts. To further illustrate the adaptability of the proposed method, we changed some conditions in the real experiment and obtained different model structures

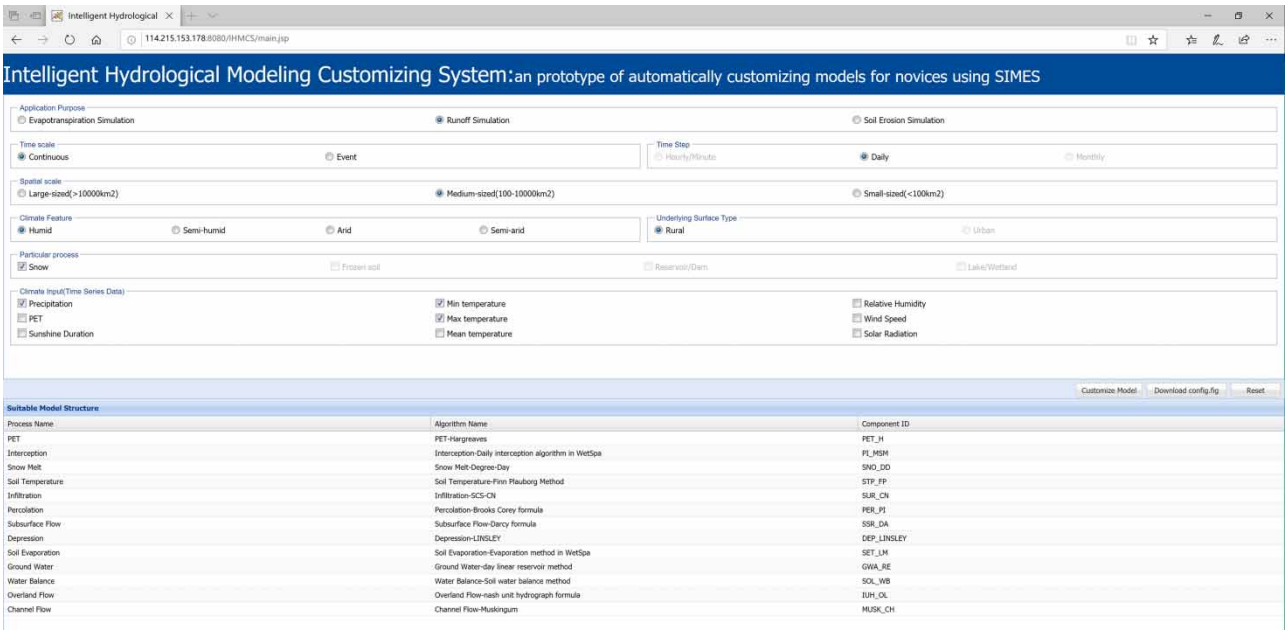

Figure 10 | Application context and model structure for the real experiment by IHMCS.

Table 5 | List of hydrological processes, algorithms, and components in the real experiment

Process	Algorithm	Component ID
PET	Hargreaves method	PET_H
Interception	Daily interception algorithm in WetSpa	PI_MSM
Snow melt	Degree-day method	SNO_DD
Soil temperature	Finn and Plauborg method	STP_FP
Infiltration	SCS-CN method	SUR_CN
Percolation	Brooks and Corey formula	PER_PI
Subsurface flow	Darcy formula	SSR_DA
Depression	Linsley method	DEP_LINSLEY
Soil evaporation	Evaporation method in WetSpa	SET_LM
Ground water	Linear reservoir method	GWA_RE
Water balance	Soil water balance method	SOL_WB
Overland flow	Nash unit hydrograph formula	IUH_OL
Channel flow	Muskingum method	MUSK_CH

accordingly. The conditions and corresponding changes of model structures are as follows:

(a) If the study area size is assumed to be small-sized rather than medium-sized and the climate type is assumed to be arid rather than humid, the subsurface flow, percolation, and ground water processes are not selected by the proposed method.
(b) If the modelling purpose is assumed to be soil erosion simulation rather than rainfall-runoff simulation, the sediment yield process is added to the model structure.
(c) If the modeller assigns the maximum temperature, minimum temperature, relative humidity, and solar radiation as the meteorological data input, the Priestley–Taylor algorithm rather than the Hargreaves algorithm is selected for the PET process.

The contrast of the model structures in the different situations is shown in Table 8.

Using this proposed method, modellers do not need to master all of the modelling knowledge, and it is easy for modellers to determine suitable models using integrated modelling frameworks. This method can reduce the modelling knowledge requirement of modellers and enable more users, especially when hydrology is not the core area of their expertise, to make use of hydrological modelling to serve their own work. With the intelligent modelling interface, modellers can use integrated modelling frameworks in a simpler and higher-level way. Users who are not familiar with modelling frameworks or components can also

Table 6 | Related knowledge and application context for process selection in the real experiment

Related knowledge	Application context	Selected processes
Process rule 1: If the purpose is rainfall-runoff simulation and the time step is daily, then infiltration/surface runoff, depression, soil evaporation, potential evapotranspiration, overland flow, channel flow, water balance, and soil temperature are selected	Purpose: rainfall-runoff simulation; Time step: daily	Infiltration/surface runoff, depression, soil evaporation, potential evapotranspiration, overland flow, channel flow, water balance, and soil temperature
Process rule 2: If the spatial scale is medium-sized and the application purpose is rainfall-runoff simulation, then percolation and ground water are selected	Spatial scale: medium-sized; Purpose: rainfall-runoff simulation	Percolation and ground water
Process rule 3: If the particular process is snow, then snowmelt is selected	Particular process: snow	Snowmelt
Process rule 4: If the underlying surface type is rural, then interception is selected	Underlying surface type: rural	Interception
Process rule 5: If the climate type is humid, then the subsurface flow is selected	Climate type: humid	Subsurface flow

Table 7 | Related knowledge and application context for algorithm identification in the real experiment

Related knowledge	Application context	Identified algorithms
Algorithm rule for the interception process: If the time step is daily, then the maximum storage method is selected	Time step: daily	Daily interception algorithm in WetSpa
Algorithm rule for the snowmelt process: If the time step is daily, then the degree-day method is selected	Time step: daily	Degree-day method
Algorithm rule for the soil temperature process: If the time step is daily, then the Finn and Plauborg method is selected	Time step: daily	Finn and Plauborg method
Algorithm rule for the infiltration process: If the time step is daily, then the SCS-CN method is selected	Time step: daily	SCS-CN method
Algorithm rule for the depression process: If the time step is daily, then the Linsley method is selected	Time step: daily	Linsley method
Algorithm rule for the potential evapotranspiration process: If the climate inputs are maximum temperature and minimum temperature, then the Hargreaves method is selected	Climate input: maximum temperature and minimum temperature	Hargreaves method
Algorithm rule for the percolation process: If the time step is daily, then the Brooks and Corey formula is selected	Time step: daily	Brooks and Corey formula
Algorithm rule for the soil evaporation process: If the time step is daily, then the Thornthwaite and Mather formula is selected	Time step: daily	Thornthwaite and Mather formula
Algorithm rule for the subsurface flow process: If the time step is daily or hourly, then the Darcy formula is selected	Time step: daily	Darcy formula
Algorithm rule for the overland flow process: If the time step is daily or hourly, then the Nash unit hydrograph formula is selected	Time step: daily	Nash unit hydrograph formula
Algorithm rule for the ground water process: If the time step is daily, then the linear reservoir method is selected	Time step: daily	Linear reservoir method
Algorithm rule for the water balance process: If the time step is daily, then the soil water balance method is selected	Time step: daily	Soil water balance method
Algorithm rule for the channel flow process: If the time step is daily or hourly, then the Muskingum method is selected	Time step: daily	Muskingum method

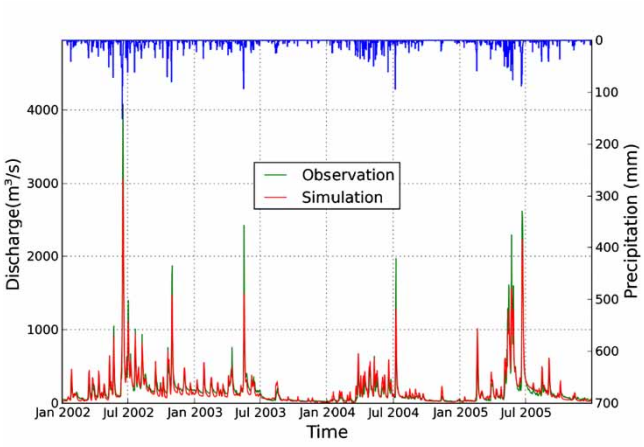

Figure 11 | Model calibration results (the Nash–Sutcliffe efficiency is 0.90).

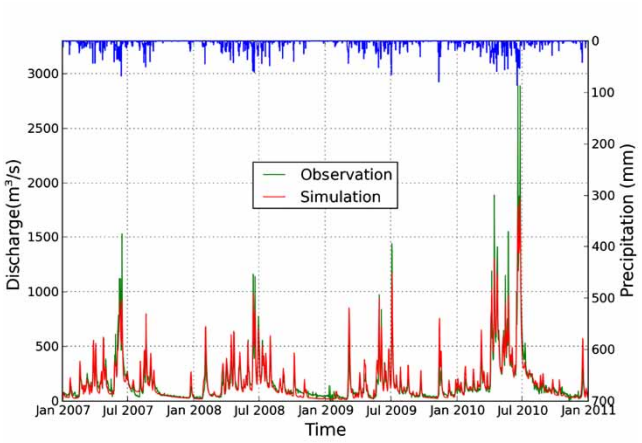

Figure 12 | Model validation results (the Nash–Sutcliffe efficiency is 0.88).

participate in modelling. The latter would be conducive to the promotion of integrated modelling frameworks. It can also help experts save the time required for modelling by providing primitive solutions for them.

Actually, the rationality of model structures depends on the quality and quantity of the knowledge base. In this study, algorithm selection is mainly constrained by time step and data availability, but other limiting factors for algorithm selection should also be considered in the future. The incompleteness of the knowledge base may cause the infeasibility of our proposed method in relatively complex environments. In addition, regional difference knowledge is very important, and it should also be included in the knowledge base.

Table 8 | Contrast of the model structures in the different situations

Process	Situation of a real experiment	Situation a	Situation b	Situation c
PET	√	√	√	√ $(\rightarrow)^{PET}$
Interception	√	√	√	√
Snow melt	√	√	√	√
Soil temperature	√	√	√	√
Infiltration	√	√	√	√
Percolation	√	○	√	√
Subsurface flow	√	○	√	√
Depression	√	√	√	√
Sediment yield	○	○	√	○
Soil evaporation	√	√	√	√
Ground water	√	○	√	√
Water balance	√	√	√	√
Overland flow	√	√	√	√
Channel flow	√	√	√	√

√: the corresponding process is selected.
○: the corresponding process is unselected.
$(\rightarrow)^{PET}$: the Priestley–Taylor algorithm rather than the Hargreaves algorithm is selected for the PET process simulation.

The knowledge base on GitHub needs to be constantly improved and expanded. Contributions to the knowledge base from the community are welcomed.

It should be noted that certain basic knowledge on hydrological modelling is always needed for hydrological modellers. For example, modellers should understand the general knowledge of hydrology (e.g. watershed conditions, how to discover and preprocess data, how to prepare parameters, and how to calibrate model parameters). Without the basic knowledge, it is still difficult to build hydrological models and interpret the simulation results properly.

CONCLUSIONS AND FUTURE WORK

In this paper, a knowledge-based method is proposed to automatically determine hydrological model structures. Specifically, the knowledge on process selection and algorithm identification is formalized in RuleML, and the knowledge on the component coupling is formalized in RDF. Then, an inference engine is implemented to generate suitable model structures according to the formalized

knowledge. Three hypothetical experiments and a real experiment are used to demonstrate how the proposed knowledge-based method can assist modellers in the determination of model structures for different application contexts. This method has the potential to reduce the modelling knowledge burden on modellers and would be conducive to the promotion of integrated modelling frameworks.

The construction of the knowledge base is a long-term project. In the future, we will continue to expand the knowledge base to enhance the feasibility of the proposed knowledge-based method in complex environments. We will attempt to develop a knowledge management platform for modellers to use to share their hydrological modelling knowledge and experience. In addition, future work should utilize other types of knowledge, such as case-based knowledge, to make hydrological modelling smarter and more robust.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (Nos. 41601423, 41601413, and 41431177), the Innovation Project of LREIS (No. O88RA20CYA), and the Outstanding Innovation Team in Colleges and Universities in Jiangsu Province. The support received by A-Xing Zhu through the Vilas Associate Award, the Hammel Faculty Fellow, and the Manasse Chair Professorship from the University of Wisconsin–Madison is greatly appreciated.

REFERENCES

Allen, R. G., Pereira, L. S., Raes, D. & Smith, M. 1998 *Crop Evapotranspiration, Guidelines for Computing Crop Water Requirements*. FAO Irrigation and Drainage Paper 56, FAO, Rome, Italy.

Anderson, J. R. 1983 *The Architecture of Cognition (Cognitive Science Series)*. Harvard University Press, Cambridge, MA, USA, pp. 35–39.

Baymani-Nezhad, M. & Han, D. 2013 Hydrological modeling using effective rainfall routed by the Muskingum method (ERM). *J. Hydroinform.* **15** (4), 1437–1455.

Boley, H., Paschke, A. & Shafiq, O. 2010 RuleML 1.0: The Overarching Specification of Web Rules. In: *4th International Web Rule Symposium*, Washington, USA, pp. 162–178.

Chau, K. W. 2007 An ontology-based knowledge management system for flow and water quality modeling. *Adv. Eng. Softw.* **38** (3), 172–181.

Cyganiak, R., Wood, D. & Lanthaler, M. 2014 *RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation*. Available from: www.w3.org/TR/2014/REC-rdf11-concepts-20140225/ (accessed 10 May 2018).

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. Modell. Softw.* **39**, 201–213.

Elag, M. & Goodall, J. L. 2013 An ontology for component-based models of water resource systems. *Water Resour. Res.* **49** (8), 5077–5091.

Furusho, C., Chancibault, K. & Andrieu, H. 2013 Adapting the coupled hydrological model ISBA-TOPMODEL to the long-term hydrological cycles of suburban rivers: evaluation and sensitivity analysis. *J. Hydrol.* **485**, 139–147.

Grimaldi, S., Petroselli, A. & Romano, N. 2013 Green-Ampt Curve-Number mixed procedure as an empirical tool for rainfall-runoff modelling in small and ungauged basins. *Hydrol. Process.* **27** (8), 1253–1264.

Hargreaves, G. H. & Samani, Z. A. 1985 Reference crop evapotranspiration from temperature. *Appl. Eng. Agric.* **1** (2), 96–99.

Harpham, Q. & Danovaro, E. 2015 Towards standard metadata to support models and interfaces in a hydro-meteorological model chain. *J. Hydrol.* **17** (2), 260–274.

Islam, A. S. & Piasecki, M. 2008 Ontology based web simulation system for hydrodynamic modeling. *Simul. Model. Pract. Theory* **16** (7), 754–767.

Jiang, P., Elag, M., Kumar, P., Peckham, S. D., Marini, L. & Rui, L. 2017 A service-oriented architecture for coupling web service models using the basic model interface (BMI). *Environ. Modell. Softw.* **92** (C), 107–118.

Lai, X., Liao, K., Feng, H. & Zhu, Q. 2016 Responses of soil water percolation to dynamic interactions among rainfall, antecedent moisture and season in a forest site. *J. Hydrol.* **540**, 565–573.

Liu, Q., Bai, Q., Kloppers, C., Fitch, P., Bai, Q., Taylor, K., Fox, P., Zednik, S., Ding, L., Terhorst, A. & McGuinness, D. 2013 An ontology-based knowledge management framework for a distributed water information system. *J. Hydroinform.* **15** (4), 1169–1188.

Liu, J., Zhu, A. X., Qin, C. Z., Wu, H. & Jiang, J. 2016 A two-level parallelization method for distributed hydrological models. *Environ. Modell. Softw.* **80**, 175–184.

Lu, B. & Piasecki, M. 2012 Community modeling systems: classification and relevance to hydrologic modeling. *J. Hydroinform.* **14** (4), 840–856.

Moore, R. V. & Tindall, C. I. 2005 An overview of the open modelling interface and environment (the OpenMI). *Environ. Sci. Policy* **8** (3), 279–286.

Morsy, M. M., Goodall, J. L., Castronova, A. M., Dash, P., Merwade, V., Sadler, J. M. & Tarboton, D. G. 2017 Design of a metadata framework for environmental models with an example hydrologic application in HydroShare. *Environ. Modell. Softw.* **93**, 13–28.

Peckham, S. D. 2014 EMELI 1.0: An Experimental Smart Modeling Framework for Automatic Coupling of Self-describing Models. In: *11th International Conference on Hydroinformatics. HIC 2014*, New York, USA.

Peckham, S. D., Hutton, E. W. H. & Norris, B. 2013 A component-based approach to integrated modeling in the geosciences: the design of CSDMS. *Comput. Geosci.* **53**, 3–12.

Qiu, L., Du, Z., Zhu, Q. & Fan, Y. 2017 An integrated flood management system based on linking environmental models and disaster-related data. *Environ. Modell. Softw.* **91**, 111–126.

Qin, C. Z., Gao, H. R., Zhu, L. J., Zhu, A. X., Liu, J. Z. & Wu, H. 2018 Spatial optimization of watershed best management practices based on slope position units. *J. Soil Water Conserv.* **73** (5), 504–517.

Rigon, R., Bertoldi, G. & Over, T. M. 2006 GEOtop: a distributed hydrological model with coupled water and energy budgets. *J. Hydrometeorol.* **7** (3), 71–388.

Škerjanec, M., Atanasova, N., Čerepnalkoski, D., Džeroski, S. & Kompare, B. 2014 Development of a knowledge library for automated watershed modeling. *Environ. Modell. Softw.* **54**, 60–72.

Thornthwaite, C. W. 1948 An approach toward a rational classification of climate. *Geogr. Rev.* **38** (1), 55–94.

Vo, N. D. & Gourbesville, P. 2016 Application of deterministic distributed hydrological model for large catchment: a case study at Vu Gia Thu Bon catchment, Vietnam. *J. Hydroinform.* **18** (5), 885–904.

Voinov, A. & Shugart, H. H. 2013 'Integronsters', integral and integrated modeling. *Environ. Modell. Softw.* **39**, 149–158.

Wang, L., Wang, Z., Liu, C., Bai, P. & Liu, X. 2018 A flexible framework hydroInformatic modeling system – HIMS. *Water* **10** (7), 962.

Ward, S., Scott Borden, D., Kabo-bah, A., Fatawu, A. N. & Mwinkom, X. F. 2019 Water resources data, models and decisions: international expert opinion on knowledge management for an uncertain but resilient future. *J. Hydroinform.* **21** (1), 32–44.

WMO 1975 *Intercomparison of Conceptual Models Used in Hydrological Forecasting*, Oper. Hydrol. Rep. 7, WMO 429. WMO, Geneva, Switzerland.

WMO 1992 *Simulated Real-Time Intercomparison of Hydrological Models*, Oper. Hydrol. Rep. 38, WMO 779. WMO, Geneva, Switzerland.

First received 27 January 2019; accepted in revised form 17 July 2019. Available online 3 September 2019