Environmental Modelling & Software 80 (2016) 245–258

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Implementation and application of a distributed hydrological model using a component-based approach

Yinping Long [a, b], Yaonan Zhang [a, *], Dawen Yang [c], Lihui Luo [a]

[a] Cold and Arid Regions Environmental and Engineering Research Institute of Chinese Academy of Sciences, Lanzhou 730000, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China
[c] State Key Laboratory of Hydroscience and Engineering, Department of Hydraulic Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 5 January 2015
Received in revised form
20 November 2015
Accepted 2 March 2016
Available online xxx

Keywords:
Component-based programming
Modelling framework
Distributed hydrological model
Runoff simulation

ABSTRACT

The application of a model to different study areas often requires that the model be modified to conform to specific characteristics, but this can be challenging due to the poor readability and reusability of the legacy codes. Component-based programming supported by a modelling framework provides a generic means to develop and modify models. This paper describes the development of a distributed hydrological model using a component-based modelling framework, which is implemented as a set of functional components that are integrated at runtime. The model was applied to runoff simulation in a large scale and data scarce alpine basin, and was further improved by incorporating a simple empirical soil freezing-thawing component. The results show that the componentised model reproduced the daily and monthly flow hydrograph with 'good' accuracy. The framework is shown to be flexible enough for model development and model modification.

1. Introduction

Hydrological models are important in facilitating our understanding of hydrological processes, performing hydrological forecasting, assessing the availability of water resources and making decisions in basin management. A hydrological model is usually developed to solve a specific problem in a specific study area, which limits its application in different regions and different spatial and temporal scales. For example, the Xinanjiang model is applicable in humid and semi-humid areas, but not in arid regions because it merely considers Dunne runoff generation mechanisms, whereas arid regions are dominated by Horton runoff (Hu et al., 2005). Consequently, for specific hydrological modelling purposes, models should be compared, selected and assessed according to the study area characteristics. In most cases (Dechmi et al., 2012; Khakbaz et al., 2011; Ouessar et al., 2009; Reyes et al., 1993), model modification is necessary when including a new process, excluding a process such as reservoir control that does not occur in a natural basin, or replacing one method with another. However, as most hydrologists are not professional programmers, model codes have poor readability and reusability, making them difficult to modify. A generic approach for coding hydrological models is needed.

Component-based programming offers a nice solution to this problem, as it emphasises the decomposition of a complex system into a set of functional components with manageable complexity (Argent, 2004; Buahin and Horsburgh, 2015; Castronova and Goodall, 2010; Peckham et al., 2013). In a component-based modelling paradigm, the components, which communicate with each other via standard interfaces, can be flexibly and rapidly assembled into a new configuration to solve a specific problem. The ease of incorporating, substituting or deleting components makes it easy to improve the model, compare methods, assess the effects of one process on the whole system and so forth. Component-based programming builds on fundamental object-oriented programming concepts, such as encapsulation, inheritance and polymorphism. A component is implemented as a class, which encapsulates some properties and methods to express a particular physical process or common functionality. The main difference between the two programming paradigms, i.e., component-based and object-oriented programming, is that component-based programming has a modelling framework to link the components together (Peckham et al., 2013). Various frameworks are in development within the water and environmental modelling domains,

* Corresponding author. Donggang West Road 320, Lanzhou, Gansu 730000, China.
E-mail addresses: longyinping11@lzb.ac.cn (Y. Long), yaonan@lzb.ac.cn (Y. Zhang).

http://dx.doi.org/10.1016/j.envsoft.2016.03.001

such as the Modular Modelling System (MMS) (Leavesley et al., 1996), the Earth System Modelling Framework (ESMF) (Hill et al., 2004), the Open Modelling Interface (OpenMI) (Moore and Tindall, 2005), the Cold Regions Hydrological Model (CRHM) platform(Pomeroy et al., 2007), the Object Modelling System (OMS) (David et al., 2013), the Community Surface Dynamics Modelling System (CSDMS) (Peckham et al., 2013), etc. These frameworks, which have defined standard interfaces for components and bring suites or libraries of components together, provide an avenue for easier modelling and can reduce the burden of repetitive coding tasks for environmental modellers (Argent et al., 2006).

However, in recent years, the environmental modelling framework studies have devoted much more attention to the enhancement of the frameworks themselves, such as analysing the software design requirements of frameworks (Whelan et al., 2014), quantifying the computational overheads introduced by adopting component-based approaches (Buahin and Horsburgh, 2015; Anthony M. Castronova and Goodall, 2013), assessing the framework invasiveness (Lloyd et al., 2011) and extending the framework functionalities (A. M. Castronova and Goodall, 2010; Formetta et al., 2014). In contrast, only a few studies have focused on framework applications in terms of contributing model components to frameworks, building models or comparing different model formulations based on frameworks (Zhou et al., 2014). The following problems may exclude potential framework users: 1) the complexity and enormity of the existing frameworks, which make it difficult to understand the interfaces and structures of frameworks and their method of operation; 2) simple application examples, often involving decomposition and recombination of a simple conceptual model, or standardisation and coupling of two or more models, which fall short of providing complete information on the frameworks and the experience of actual use; and 3) the lack of available framework-based model components. Thus, in addition to improving framework performances, more application examples are required to boost potential users. The objectives of this paper are to present a complete example of implementation and application of a distributed hydrological model based on a component-based modelling framework and to contribute it to the framework component base. The major challenges of this work lie in understanding the mechanism of hydrological processes and decomposing the hydrologic system into components at the optimum level of granularity.

The framework adopted for model implementation and simulation in this study is known as the Heihe river basin Open Modelling Environment (HOME). The Cold and Arid Regions Environmental and Engineering Research Institute of the Chinese Academy of Sciences, with the support of the National Natural Science Foundation of China, has been developing the HOME framework since 2011 to serve as an integrated platform that supports model development, coupling, management, simulation and analysis for 'Integrated Research on the Eco-hydrological Processes of the Heihe Basin'. HOME uses object-oriented techniques based on JAVA to enable modellers to customise their models by coupling individual components that express different processes. It comprises four parts: a kernel system, a graphic modelling tool, a component base and a database. So far, the kernel system that supports component development, component coupling, model configuration and execution has been realised and can run on any operating system without the other three parts. Our work was based on the kernel system of HOME.

We focus on the runoff simulation of the Buha River basin, which is the largest tributary of the largest inland salt-water lake in China, Qinghai Lake, located in the north-eastern Tibetan Plateau. The water level of the lake was reported to have decreased by about 3.7 m overall from 1959 to 2004, but to have increased by nearly 1 m from 2004 to 2009 (Zhang et al., 2014). Modelling the discharge of the Buha River, which accounts for almost half of the lake's inflow, is hence important to assess the water resources of the lake. However, few data are available for this alpine basin, which covers a large area of approximately 15,000 km^2, with one hydrometric station at the outlet and seven meteorological stations in and around the area, making it quite a challenge to simulate the hydrological processes.

Thus, based on the HOME framework, the Geomorphology-Based Hydrological Model (GBHM) initially developed by Dawen Yang (Yang et al., 2001b) in Fortran, was implemented. This distributed hydrological model (Xu et al., 2008; Xu et al., 2013; Yang et al., 2001a) can be applied to large, complex basins because it adopts the flow interval-hillslope discretisation scheme, is able to represent spatial variability such as topography, land cover and soil type and incorporates vegetation dynamics using the temporal leaf area index (LAI). In particular, this physically-based distributed hydrological model is quite useful in data-poor environments because it implements very few empirical parameters and does not quite rely on the observed data for parameter calibration. Nevertheless, the original code was not well modularised and was difficult to reuse, as the input and configuration parts needed to be modified for different study areas. Additionally, it did not consider the effects of the soil freezing-thawing process, which significantly affects the hydrological processes in the Buha River basin. Consequently, the original GBHM was componentised and a simple empirical soil freezing-thawing component was incorporated.

In the following sections, the HOME framework is presented and the reconstruction of the distributed hydrological model based on the HOME is described in detail. The hydrological simulation of the Buha River basin based on the new model and the analysis of the results are then specified. Finally, we summarise the findings, present our conclusions and outline future work.

2. Heihe river basin Open Modelling Environment (HOME)

HOME is an integrated framework developed specifically for eco-hydrological model development. Its aims are to 1) enhance the development and evaluation of scientific components; 2) facilitate model development and improvement through the use of well-componentised legacy codes; 3) provide standard interfaces for model importing and coupling for solving complex problems; and 4) provide a wide range of analysis components. To achieve these aims, HOME is implemented in four parts: a graphic modelling tool, a database, a component base and a kernel system (Fig. 1). The graphic modelling tool is mainly used to assist the model configuration, control the model simulation and visualise the modelling outputs. The database and component base provide modelling resources. The physical process components and data processing and analysis tools are managed by the component base. The kernel system, which as previously mentioned is able to run independently, requires at least three parts to work: a configuration file, a model entity and a runtime. The model comprises a set of components, contexts and data pools, which are the key concepts used by HOME. As only the kernel system was used for the model implementation and simulation, only the key elements of the kernel system are introduced in detail in this section.

2.1. Key HOME entities

2.1.1. Component

The term 'component' in the HOME refers to a modelling entity that implements one of the eco-hydrological processes or a data processing method. Each component extends an abstract class and overrides its three methods, init, run and clear, to represent the