Advances in Engineering Software, Vol. 37, No. 3, 2006, pp. 146-158

A Web-based flood forecasting system for Shuangpai Region

Xiang-Yang Li[1], K.W. Chau[*2], Chun-Tian Cheng[1], Y.S. Li[2]

[1]Department of Civil Engineering, Dalian University of Technology, Dalian, 116024, P.R. China
[2]Department of Civil and Structural Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract: Traditional flood forecasting and operation of reservoirs in China are based on manual calculations by hydrologists or through standalone computer programs. The main drawbacks of these methods are shortage of prediction time due to time-consuming nature, individual knowledge, lack of communication, absence of experts, etc. A Web-based flood forecasting system (WFFS), which includes five main modules: real-time rainfall data conversion, model-driven hydrologic forecasting, model calibration, precipitation forecasting, and flood analysis, is presented in this paper. The WFFS brings significant convenience to personnel engaged in flood forecasting and control and allows real-time contribution of a wide range of experts at other spatial locations in times of emergency. The conceptual framework and detailed components of the proposed WFFS, which employs a multi-tiered architecture, are illustrated. Multi-tiered architecture offers great flexibility, portability, reusability and reliability. The prototype WFFS has been developed in Java programming language and applied in Shuangpai region with a satisfactory result.

Keywords: Flood forecasting; Web; Multi-tiered; Rainfall-runoff model; Component.

1. Introduction

The effective and efficient operation of reservoirs relies on accurate and prompt forecast of reservoir inflows. Traditional reservoir forecasting in China is normally based on manual calculations by experienced hydrologists or through standalone computer programs. Owing to the complexity of hydrological process, manual calculation is not only difficult to apply, in particular to novice engineers, but also a very time-consuming task even for specialist hydrologists. The emergence of standalone computer programs on flood forecasting system (FFS) greatly improved the forecasting level and has been applied well throughout the past decade. In general, these systems have realized many advantages in terms of productivity such as easiness of operating, great speed of processing, and reduced possibility of error due to the subjective judgment involved. For example, the flood-warning decision-support system for Sacramento, California (Ford, 2001) increased warning lead time, thus providing an opportunity to reduce damage and save lives; Koussis et al. (2003) presented a system of flood forecasts for urban basin with integrated hydro-meteorological model, which was proved to be a promising tool for forecasting flood risk in the Kifissos basin, Athens. However, with the recent advent of integrated water resources management system, shortfalls about these standalone systems emerge: local information cannot be shared directly by other geographically dispersed departments or individuals who need it; departments in different roles cannot collaborate in flood forecasting due to the lack of communication infrastructure within the system; productivity is hampered by the lack of effective channels for prompt

---

[*] Corresponding author

information access and collaboration among project personnel (Liu et al., 2003), etc. A Web-based, distributed computing environment is highly desirable in modern flood prediction and control (FPC). The recent advances in Internet technology and distributed computing paved the way for developing Web-based distributed systems (Xue and Xu, 2003), which is a trend of modern software systems.

Web-based technology, which is a powerful tool for communication in itself, provides a convenient and cost-effective manner for gathering, filtering, managing, and sharing information (Ng et al., 2003). There are many Web-based applications developed for various design and decision-making problems, such as architectural design and performance evaluation (Goedicke and Meyer, 1999; Molenaar and Songer, 2001), collaborative research and development (Tamine and Dillmann, 2003), and mechanical design and manufacture (Xue and Xu, 2003; Xu and Liu, 2003), the chronology of British hydrological events (Black and Law, 2004). However, there is little research on Web-based flood forecasting systems (WFFS). Until recently, Al-Sabhan et al. (2002) discussed tremendous potential of the Web technology on the application of real-time watershed analysis and presented a real-time hydrological model for flood prediction using GIS and the WWW. The Web-based environment is emerging as a very important development and delivery platform for real-time flood forecasting system.

In this paper, a framework of WFFS, which employs a multi-tiered architecture, is presented and implemented. The WFFS brings significant convenience to personnel engaged in flood forecasting and control and allows real-time contribution of a wide range of experts at other spatial locations in times of emergency so long as Internet is accessible, and therefore provides opportunities for improving the transfer of information and knowledge from the hydrological scientists and managers to decision makers. Using Web browsers can also eliminate the need to install and maintain specialized “client” software on each machine. Hence the overall cost of system implementation, distribution and administration is significantly reduced (Xu and Liu, 2003).

The objective of the present study is to develop a WFFS by modeling the entire flood forecasting process and database using the Java programming language tailor-made to ambient conditions in China. The paper is organized as follows. A multi-tiered architecture of the Web-based distributed system is briefly described in Section 2. The implementation of the WFFS is presented in Section 3. A case study example is presented in Section 4 that illustrates in detail the application of the WFFS. Finally, conclusions are drawn in Section 5.

2. Multi-tiered architecture of the Web-based distributed system

Because of the differences in the field context of different FPC activities in different catchment regions, most FFSs have to be customized for specific FPC activities. Unfortunately, developing a new system is expensive in terms of money, time, and effort. Moreover, such systems are more error prone and very expensive to maintain (Johnson and Foote, 1988). A multi-tiered architecture of the Web-based distributed system described in this paper provides a solution to development of WFFS by capturing the common components presented in several FFSs. Thus the development time can be greatly reduced since most of the common elements have been implemented.

The multi-tiered architecture of the Web-based distributed system is shown in Fig. 1. In this architecture, different independent systems and databases can be distributed at different locations, represented as nodes such as *WFFS*, *Flood Operating System (FOS)* and *Information Service System* (ISS), among which *WFFS* is focused in this paper. Each node is composed of three tiers: Web Server (presentation tier); Application Server (application tier); and, Distributed Database (data tier). Each tier has a different responsibility in the overall deployment, and within each tier there can be one or more components. It should be noted that these tiers are purely abstractions, and they may not correspond to physical distribution.

A presentation tier contains components dealing with user interfaces and user interaction. For example, Java Servlets, Java Server Pages (JSP), and/or Java Applets could be employed for the presentation tier of a Web-based deployment.

An application tier contains components that work together to solve business logic, such as *Conversion*, *Parameter Calibration*, and *Hydrologic Forecasting*.

A data tier is used by business logic tier as the repository of the entire system.

On the front line of the Web site are the Web Servers that act as the presentation tier. Web Servers dynamically format content as HTML, Applet or JSP to be displayed by Web browsers. All the business logic of the system resides in the Application Servers that act as application tier. Application Servers receive requests from Web Servers, look up information in databases and process the requests. The processed information is then passed back to the Web Servers where it is formatted and displayed. The relational database management system is the repository of the entire system. Data including model parameters and real-time and/or historical hydrological data of reservoir and/or river are stored in databases. A layered system is a well-designed system because each layer is responsible for a separate task. The separation of application tier from presentation tier makes it possible to develop multiple presentations for the same process for different clients, or to change the presentation of a forecasting process without the necessity to modify the code that implements the forecasting process. More user-friendly Web-based interfaces can thus be designed. It is similarly possible to plug in a different set of business rule component implementations within the application tier, or to plug in a different database in the data tier, with relatively minor effects on the other tiers. Details of the advantages of three-tiered/multi-tiered architecture can be referred to Matena et al. (2003) and Xu and Liu (2003).

3. Implementation of the WFFS

The conceptual framework of WFFS is shown in Fig. 2, using the architecture presented as above. General users, authorized users, experts in hydrology, system administrators, etc. can synchronously access the Web-based interfaces independently, for inputting data, forecasting, operating, analyzing, making decision, querying, etc. The requests received in the presentation tier are proposed to the application tier. Components residing in the application tier constitute five main modules, real-time rainfall data *conversion*, *hydrologic forecasting* using models (i.e. rainfall-runoff models), *model parameter calibration*, *precipitation forecasting*, and *flood analysis*. Each module is accomplished by the incorporation of a number of well-defined components, which perform independent tasks. Components running inside the Application Servers are Enterprise JavaBeans (EJB) components based on distributed objects. A distributed object is an object that is callable from a remote system. It can be called from an in-process client, an out-of-process client, or a client located elsewhere on the network. Clients can achieve distributed objects using many technologies, including the OMG's CORBA, Microsoft's DCOM, and Sun's Java RMI-IIOP. Java RMI-IIOP (which stands for Java Remote Method Invocation over the Internet Inter-ORB Protocol) is EJB's de facto mechanism for performing simple, powerful networking. RMI-IIOP use Java Naming and Directory Interface (JNDI) to locate distributed objects. JNDI, the standard way of looking things up over the network, provides a standard interface for locating users, machines, objects, and services. See Appendix A, B of Matena et al., (2003) for details. Fig. 3 shows how a client can call a distributed object. The following is an explanation of the diagram:

1. The client calls a stub, which is a client-side proxy object. This stub is responsible for masking network communications from the client. The stub knows how to call over the network using sockets, massaging parameters as necessary into their network representation.
2. The stub calls over the network to a skeleton, which is a server-side proxy object. The skeleton masks network communication from the distributed object. The skeleton understands how to receive calls on a socket. It also knows how to massage parameters from their network representations to their Java representations.
3. The skeleton delegates the call to the distributed object. The distributed object does its work, and then returns control to the skeleton, which returns to the stub, which then returns control to the client.
4. The request interceptor intercepts requests from the client, performs the middleware that distributed object needs, and then delegates the call to the distributed object. Complicated middleware services, such as resource pooling, networking, security, multithreading, clustering and distributed computing, are provided in Application Server. Examples of such Application Server products are BEA's WebLogic, iPlanet's iPlanet Application Server, IBM's WebSphere.

A schematic diagram of the process for flood forecasting is shown in Fig. 4. Real-time rainfalls and water levels are first measured by remote sensing facilities (RSF) in hydrological stations dispersed throughout the region. Then the sequences that generally have 1h temporal interval are transmitted and stored in the *Remote Database* (see Table 1). However, they cannot be applied in rainfall-runoff models directly because the real-time hydrometeorological data does not have the same temporal interval as those for rainfall-runoff models. The *conversion* module is designed to convert remote sensing hydrologic data to sequences in a uniform interval and to store them in corresponding tables in the *Common Database*. The table structure is similar to the one in the *Remote Database*, with the exception of the time intervals. In real time simulation, the *conversion* module runs in the background, periodically acquiring data from the *Remote Database*, converting and storing it into the *Common Database*. This data provides input for the *model parameter calibration* module and the *hydrologic forecasting* module that consists of a runoff generation sub-module and a runoff routing sub-module. With appropriate rainfalls and other spatial hydrologic data, different runoff generation methods are employed to transform precipitation into runoff for each sub-area. The outflow hydrograph from each sub-area is then routed down the channels to the main outlet by some runoff routing methods such as Muskingum method (Franchini and Lamberti, 1994). After hydrologic forecasting is finished, forecasted flow sequences are passed to *flood analysis* module for further analysis. Characteristics and frequency of the flood are analyzed within the *flood analysis* module and stored into the database together with flow sequences. The main task of the m*odel parameter calibration* module is to calibrate and verify model parameters used in the *hydrologic forecasting* module by applying historical floods.

The following is an example of client code that invokes methods on a distributed object: Rainfall_Runoff. It’s a simple representation of the real-time hydrologic forecasting process.

```
/*
* Setup properties for JNDI initialization.
* These properties will be read-in from the command-line.
*/
Properties props = System.getProperties();
/*
* Obtain the JNDI initial context.
```

```
* The initial context is a starting point for connecting to a JNDI tree. We choose our JNDI
* driver, the network location of the server, etc. by passing in the environment properties.
*/
Context ctx = new InitialContext(props);
/*
* Get a reference to the home object – the factory for Rainfall_Runoff EJB Objects
*/
Object obj = ctx.lookup("Rainfall_Runoff ");
/*
* Home objects are RMI-IIOP objects, and so they must be cast into RMI-IIOP objects
* using a special RMI-IIOP cast.
*/
Rainfall_RunoffHome home = (Rainfall_RunoffHome)
javax.rmi.PortableRemoteObject.narrow(obj, Rainfall_RunoffHome.class);
/*
* Use the factory to create the EJB Object, a proxy object of the distributed object.
*/
Rainfall_Runoff model = home.create(parameters);
/*
* Call business methods on the EJB object. The EJB object will delegate the call to the
* distributed object, receive the result, and return it to us.
* See Table 2 for descriptions of the methods defined in Rainfall_Runoff interface.
*/
model.init();
model.retrieve();
model.averPrecipitation();
model.runoff();
model.routing();
// Get the result outflow hydrograph of the main outlet.
Outflow outflow = model.getOutflow();
model.remove();
```

Rainfall-Runoff Model

Rainfall-runoff models play a key role in *hydrologic forecasting* module. Many of them are developed based on conceptual representations of the physical process of the water flow lumped over the entire catchment area. They apply well under their specific environment. Examples of this type of models are the Sacramento model (Burnash, 1995), the Tank model (Sugawara, 1978), The HBV model (Bergstrom, 1995), Dahuofang model, and the Xinanjiang model (Zhao et al., 1980; Zhao, 1992). Lettenmaier and Wood (1993) provide a survey of short-term forecasting models, and a variety of computer programs are available to implement these. Each model is encapsulated in a component in this study. A component is a code that implements a set of well-defined interfaces. It is a manageable, discrete chunk of logic. Components are not entire applications - they cannot run alone. Rather, they can be used as puzzle pieces to solve some larger problem. They provide the following benefits: simplicity, portability, flexibility, component reusability, and ability to build complex applications. In order to achieve such purpose, the structure of *Rainfall-Runoff Model* Components must be designed properly. Hence, an identical interface named *Rainfall_Runoff*, which is exposed to other program modules, is abstracted from all rainfall-runoff models. Different *Rainfall-Runoff Model* Components inherit and implement this interface (shown in Fig. 5) with different algorithms. From the point of view of application developers, the manner how the *Rainfall_Runoff* interface is implemented in different components does not matter and they only need to program with the interface. In this way *Rainfall-Runoff Model* Components can be easily reused and assembled into other WFSSs without additional work. Table 2 shows the names and descriptions of main methods defined in the identical interface.

Xinanjiang model, which is applied in the case study, is taken as an example to explain the whole process of rainfall runoff simulation. The Xinanjiang model has been successfully and widely applied in humid and semi-humid regions in China since its development in the 1970s. Details about it can be referred to Zhao (1992). It is encapsulated in this FFS as a component named *XinanjiangModel* Component (Fig. 5), which implements the *Rainfall_Runoff* interface. When a *XinanjiangModel* Component instance is to be initialized, parameters like RSTCD and SUBBASIN of the area are provided to the method init(). This method is responsible for retrieving related record of parameters from table XAJ (shown in Table 3) in the *Common Database*. The method retrieve() retrieves related data such as real-time or historical rainfall data and initial values of antecedent soil moisture (ASM) from the *Common Database*. The method averPrecipitation() computes mean areal precipitation hyetograph by weighting hyetographs of rain gauges distributed in each sub-area. The method runoff() and routing() operate together to transform rainfall data into runoff at the main outlet by using Xinanjiang model.

Other conceptual rainfall-runoff models such as Sacramento model, Tank model, and Dahuofang model can be implemented in the same way as Xinanjiang model, except that different algorithms for rainfall-runoff simulation are employed. Each of them is particularly suitable under a specific hydrological and geologic environment. Moreover, it is very easy to extend the FFS. When a new rainfall-runoff model is to be integrated to the system, developers only need to code a new *Rainfall-Runoff Model* Component, which implements the *Rainfall_Runoff* interface with the corresponding algorithm. The tedious and error prone modifications to the program skeleton can thus be avoided.

Flood Analysis

Peak discharge, peak time, and total runoff volume are three important characteristics to evaluate a specific flood and they provide foundation for the operation of reservoirs. These characteristic values are calculated within *flood analysis* module and stored in the database for the subsequent process. The flood frequency is also calculated by comparing the characteristics with the designed flood. There are two choices of table structure available for defining flood characteristics. Fig. 6(a) and Fig. 6(b) show their similar Entity Relationship Diagram (ERD) and Fig. 6(c) lists the descriptions of all fields. Table *Characteristics* stores characteristics of a specific alternative and has a one-to-many (1:n) relationship with table *FlowSequence,* which represents flow sequence of the alternative. If the table *Characteristics* shown in Fig. 6(a) is adopted, the table *FlowSequence* will be very cumbersome. Thus, a primary key ID of the table *Characteristics* is defined (Fig. 6(b)), which acts as an identifier of an alternative. In Fig. 6(a), the fields: RSTCD; FLOOD; and, NO of table *FlowSequence* now can be replaced with an identifier ID (Fig. 6(b)). Thereby, the storage and complexity are greatly reduced. Each record of the Fig. 6(b) represents an alternative made by an authorized user with a specific hydrologic model. The user, the hydrologic model, peak discharge, peak time, and total runoff volume of each alternative are recorded in table *Characteristics*. When alternatives made by a specific user are evaluated, the values of field ID related to the user can be found by applying some SQL query on the table *Characteristics*. Then details of alternatives related to field ID such as flow sequence shown in Fig. 6(b) can be obtained. By the definitions of table structures, WFFS can distinguish the forecasted results among multiple users and thus implement the collaborative process because all users can query the forecasted results by browsing through the result pages.

Parameter Calibration

Rainfall-runoff models generally have a large number of parameters, which cannot be obtained directly from measurable quantities of catchment characteristics, and hence model calibration is entailed. The objective of model calibration is to determine a set of parameters through application to historical hydrological data, such that the model simulates the hydrological behavior of the catchment as closely as possible.

Many literature applied optimization methods in solving the automatic calibration of rainfall-runoff model (Madsen, 2000; Cheng et al., 2002). Different calibration methods differ in the objectives and automatic optimization algorithms employed. Each method can be implemented in an independent component, which can be invoked without any difference. The method proposed by Cheng et al. (2002) has been proved to have good performance through practice on Shuangpai Reservoir. It combined a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall–runoff model calibration problem. It is implemented in the *Parameter Calibration* component as an isolated component in the WFFS. A Web-based interactive interface of parameter calibration for Xinanjiang model is shown in Fig. 7. Hydrological users can perform calibration by adjusting initial ranges of the parameters, choosing representative historical floods and analyzing through Web browsers. These model parameters can be further calibrated and updated to simulate the hydrologic process better using new flooding data.

Precipitation forecasting

If rainfall is measured on the ground for flood forecasting, the hydrologic response time is only a few hours, which are sometimes inadequate. There is little point in updating the flow model in real time by incorporating measurements (Hoos et al., 1989). It is, however, feasible to extend the preparation time of a civil protection authority by forecasting the rainfall. Using the rainfall forecast as input to a rainfall-runoff model, the runoff and outflow hydrograph can be forecasted with an increased forecast lead time (FLT), which can be practical and useful. Integrated hydrometeorological model approaches have been tried for forecasting rainfall in the context of hydrologic applications, i.e., for the purpose of increasing FLT (Anderson et al., 2002; Koussis et al., 2003). Anderson et al. (2002) integrated an Eta Model, a mesoscale model, MM5, and a rainfall-runoff model, HEC-HMS, to transform precipitation forecasts into runoff forecasts with a 48-h lead time. But the process cannot be applicable in China since precipitation in grid form is unavailable in many regions. Therefore a coarser method is adopted, which need manual input of precipitation data according to daily weather forecast. It is useful for general flood forecasting and reservoir operation. Usually, the maximum lead time of the future precipitation input depends on the flood status and available hydrometeorological data. For a large scale basin with a water holding capacity of more than 100 million cubic meters, the routing time is usually more than 10 hours or even longer. Therefore, a maximum lead time of 48-h of the future precipitation input is enough because total FLT of about 60 hours is sufficient for flood control operation in the future. An interface named *Precipitation_Forecast* (shown in Table 4) is preserved for future extension. When the precipitation forecast is applicable, a new component would be developed to implement the *Precipitation_Forecast* interface and deployed into an application server.

Exception Handling

Exception handling module designed to detect data integrity and validity is integrated in each module of the WFFS. The main exceptions defined in the WFFS and their descriptions are listed in Table 5. They are all subclass of Exception which is included in java basic class libraries. Whenever an exception/error occurs during the forecasting process, the exception handling module catches it, identifies the type of exception and switches to corresponding statements branch. Generally, a user-friendly interface represented as web page is prompted to users indicating the probable reasons and corresponding solutions. For example, if there is no real-time precipitation data in the database, the exception handling module will prompt a web page indicating the problem and direct users to input precipitation data through corresponding interactive web pages. The following exception handling statements show sample solutions.

```
if (exception instanceof SystemException) {
        //Tell users that some system error occurred and the program will be terminated.
}else if (exception instanceof UnAuthorizedException) {
        //Direct users input valid user id and password.
}else if (exception instanceof ASMException) {
        //Direct users input ASM data through corresponding interactive web pages.
}else if (exception instanceof PeriodRainException) {
        //Direct users input precipitation data through corresponding interactive web pages.
}…
```

4. Application Example

The WFFS is applied to the Shuangpai Reservoir (Fig. 8). The reservoir, with a drainage area of 10,594 km^2 and a water holding capacity of up to 373.8 million cubic meters, is located in Hunan province of the southern China and at the downstream of the Xiaoshui Stream, which is one of tributary rivers in the Xiangjiang River. The length of the main stream is 154.9 km with an averaged slope of 0.61%. The area is in subtropical monsoon zone with rich rainfalls and good vegetation cover. The annual rainfall is 1500 mm; the averaged depth of runoff is 893 mm and the averaged discharge is 300 m^3/s. However, the temporal distribution of the rainfall during a given year is significantly heterogeneous in this area. The flood events in this area are mainly due to the thunderstorms. 45.9% of the total rainfall falls between April and June, and 34% of the total rainfall between September and October, which are referred to as the high flow periods. The reservoir is used for power generation, flood control, as well as for irrigation purposes. Table 6 summarizes the rain gauge stations covered in this study, including the representing area and the corresponding weighting for each rain gauge.

Xinanjiang model with 3h temporal interval is adopted as a result of its satisfactory previous long-term practice on Shuangpai Reservoir. The region is divided into 12 sub-areas, each of which has the same set of model parameters. Each sub-area is represented by a rain gauge station. Table 7 shows the adopted model parameters, which are acquired by parameters calibration introduced in the last section. The flood event occurred on Aug 30, 1999 is selected to illustrate the application of the WFFS. The web page used for initial setting is shown in Fig. 9. The system presents two hydrological models, Xinanjiang model and Dahuofang model, both of which are widely used in China. There are two choices of Data Source entry: *Automated*; and, *Manual*. *Automated* option allows hydrologic data to be determined within the WFFS through conversion and *Manual* option requires users to input data manually. In general flood forecasting process *Automated* option is selected.

It is well known that a flood forecasting system is a crucial component in flood mitigation and also is a complex process. For certain important large-scale reservoirs, cooperation and communication among federal, state, and local stakeholders is required when heavy flood events are encountered. No single agency is authorized to give flood-forecasting result. The final disseminated result is usually a bargaining solution compromised by different parties and experts. In this example, experts in hydrology can adjust empirical parameters and initial conditions of hydrologic models whose values vary from a forecasting time to the next according to hydrometeorological information and experiences. For example, Fig. 10 provides an interactive web page for hydrologists adjust ASM for a certain flood.

The graphs of the observed inflow into Shuangpai Reservoir in comparison to two Xinanjiang model simulated inflows are shown in Fig. 11. It is noted that alternative 1 can simulate the flood with good agreement of peak timing and peak flow while alternative 2 differs significantly in peak flow. This is because the initial values of ASM employed differ significantly and those employed by alternative 1 represent ASM better. By this means, different hydrological models can be applied to the same region and even the same model can be applied repeatedly by adjusting some empirical parameters such as ASM by users independently until a satisfactory result is accomplished in accordance with their experiences. A summary of all alternatives made by different authorized users is listed in Fig. 12. By referring to experts' remarks, decision makers can choose an ideal alternative among them after synthetically analysis, prior to arriving at a final decision of reservoir operation. The WFFS provides novice engineers, specialist hydrologists and decision makers a platform to communicate to each other on-line. Thus, the widespread participation from large diverse groups of individuals and experts within the flood control management agencies and research institutions is a significant factor in enhancing forecasting accuracy and reliability.

5. Conclusions

This research presented a Web-based distributed system for real-time flood forecasting which has been applied on Shuangpai Reservoir with a satisfactory result. This system has four significant features - wide accessibility, flexibility, reusability, and user friendliness. Simple and inexpensive Web browsers can expand the use of forecasting system to many people in the domain without additional networking facilities. This permits the local authorities to take advantages of the knowledge and experience of hydrology experts, even if those experts are not in the vicinity. They can visit the website and make decisions through web browsers. Therefore, WFFS provides opportunities for streamlining the transfer of information and knowledge from the hydrological scientists and managers to decision makers and thus enhancing forecast accuracy and reliability. Furthermore, EJB components that encapsulate business logic such as real-time hydrologic forecasting and model calibration can be easily reused and deployed in distributed multi-tier environment. The clients of the component can deal with only a single exposed component interface, without paying regard to the component's composition and implementation. Thus, this is a powerful, flexible model, which can be easily extended as required. By capturing characteristics of hydrologic model and calibration algorithm and implementing them in EJB components, great reusability can be achieved. The WFFS presented in this paper renders flood forecasting process easier and more convenient through web browsers.

Acknowledgement

This research was supported by the Internal Competitive Research Grant of Hong Kong Polytechnic University (A-PE26) and the Natural Science Foundation of Liaoning Province, China (No. 20032114).

References

Al-Sabhan, W., Mulligan, M., Blackburn, G.A. (2002). “A real-time hydrological model for flood prediction using GIS and the WWW.” *Computers, Environment and Urban Systems*, 27(1), 9-32.

Anderson, M.L., Chen, Z.-Q., Kavvas, M.L., Arlen, F. (2002). “Coupling HEC-HMS with atmospheric models for prediction of watershed runoff.” *Journal of Hydrologic Engineering, ASCE*, 7(4), 312-318.

Bergstrom, S. (1995). The HBV model, Computer Models of Watershed Hydrology (ed. Singh, V.P.), Water Resources Publications, Colorado, 443-476.

Black, A.R. & Law, F.M. (2004). “Development and utilization of a national web-based chronology of hydrological events.” *Hydrological Sciences Journal*, 49(2), 237-246.

Burnash, R.J.C. (1995). “The NWS river forecast system — catchment modeling.” In: Singh, V.P. (Ed.). *Computer Models of Watershed Hydrology*, Water Resources Publications, Colorado, pp. 311–366.

Cheng, C.T., Ou, C.P., Chau, K.W. (2002). “Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall–runoff model calibration.” *Journal of Hydrology*, 268(1-4), 72-86.

Ford, D.T. (2001). “Flood-warning decision-support system for Sacramento, California.” *Journal of Water Resources Planning and Management, ASCE*, 127(4), 254-260.

Franchini, M., and Lamberti, P. (1994). “A flood routing Muskingum type simulation model based on level data alone.” *Water Resources Research*, 30(7), 2183–2196.

Goedicke, M., Meyer, T. (1999). “Web-based tool support for dynamic management of distribution and parallelism in integrating architectural design and performance evaluation.” *Proc., Int. Workshop on Software Engineering for Parallel and Distributed System*s, IEEE, New York, 156–163.

Hoos, A.B., Koussis, A.D., and Beale, G.O. (1989). “A channel dynamics model for real-time flood forecasting.” *Water Resour. Res.*, 25(4), 691–705.

Johnson, R.E., Foote, B. (1988). “Designing reusable classes.” *J. Object-Oriented Program,* 1(2), 22–35.

Koussis, A.D., Lagouvardos, K., Mazi, K., Kotroni, V., Sitzmann, D., Lang, J., Zaiss, H., Buzzi, A., Malguzzi, P. (2003). “Flood forecasts for urban basin with integrated hydro-meteorological model.” *Journal of Hydrologic Engineering, ASCE,* 8(1), 1-11.

Lettenmaier, D.P., Wood, E.F. (1993). “Hydrologic forecasting.” *Handbook of Hydrology*, D.R. Maidment, ed., McGraw-Hill, New York.

Liu, D., Cheng, J.X., Law K.H., Wiederhold G., Sriram R.D. (2003). “Engineering information service infrastructure for ubiquitous computing.” *Journal of Computing in Civil*

Engineering, ASCE, 17(4), 219-229.

Madsen, H. (2000). “Automatic calibration of a conceptual rainfall–runoff model using multiple objectives.” *Journal of Hydrology*, 235(3-4), 276–288.

Matena, V., Krishnan, S., DeMichiel, L., Stearns, B. (2003). *Applying Enterprise JavaBeans™: Component-Based Development for the J2EE™ Platform* , Second Edition. Addison-Wesley, 2003.

Molenaar, K.R., Songer, A.D. (2001). “Web-based decision support systems: case study in project delivery.” *Journal of Computing in Civil Engineering, ASCE*, 15(4), 259-267.

Ng, S.T., Palaneeswaran E., Kumaraswamy M.M., (2003). “Web-based centralized multiclient cooperative contractor registration system.” *Journal of Computing in Civil Engineering, ASCE*, 17(1), 28-37.

Sugawara, M., (1978). *Automatic calibration of the tank model*, Hydrologic Sciences Bulletin, 24(3).

Tamine, O., Dillmann, R. (2003). “KaViDo - a web-based system for collaborative research and development processes.” *Computers in Industry*, 52 (1), 29-45.

Xu, X.W., Liu, T. (2003). “A web-enabled PDM system in a collaborative design environment.” *Robotics and Computer-Integrated Manufacturing*, 19(4), 315–328.

Xue, D., Xu, Y. (2003). “Web-based distributed system and database modeling for concurrent design.” *Computer-Aided Design*, 35(5), 433-452.

Zhao, R.J., Zhang, Y.L., Fang, L.R. (1980). “The Xinanjiang model.” *Hydrological Forecasting Proceedings Oxford Symposium*, IASH 129, 351–356.

Zhao, R.J. (1992). “The Xinanjiang model applied in China.” *Journal of Hydrology*, 135(1-4), 371–381.

Table 1. The table structure of hourly precipitation in *Remote Database*

Field Name	**Key**	**Type**	**Description**
STCD	Y	CHAR	The code of rain gauges distributed in the region
DT	Y	DATE	Time in 1-hour intervals
P		NUMBER	Precipitation corresponding to the DT field

Table 2. Descriptions of methods in *Rainfall_Runoff* interface

Method Name	**Description**
init()	Initializing the *Rainfall-Runoff Model* component to make it ready to be invoked for forecast simulation.
retrieve()	Retrieving historical or real-time hydrological data such as precipitation sequences, initial values of Antecedent Soil Moisture.
averPrecipitation()	Calculating the depth and time distribution of mean areal precipitation (MAP) for each sub-area.
runoff()	Translating precipitation into runoff for sub-area by using a certain rainfall-runoff model with its appropriate parameters.
routing()	Routing the outflow hydrograph from each sub-area down the channels to the main outlet by some routing method such as Muskingum method.
getOutflow()	Getting the result outflow hydrograph of the main outlet.

Table 3. The table structure of parameters of Xinanjiang model

Field Name	**Type**	**Key**	**Description**
RSTCD	CHAR	Y	The code of the river or reservoir
SUBBASIN	INTEGER	Y	The sub-basin Number in the region
Um (mm)	Number		Averaged soil moisture storage capacity of the upper layer
Lm (mm)	Number		Averaged soil moisture storage capacity of the lower layer
Dm (mm)	Number		Averaged soil moisture storage capacity of the deep layer
B	Number		Exponential parameter with a single parabolic curve, which represents the non-uniformity of the spatial distribution of the soil moisture storage capacity over the catchment.
Im (%)	Number		Percentage of impervious and saturated areas in the catchment.
K	Number		Ratio of potential evapotranspiration to pan evaporation.
C	Number		Coefficient of the deep layer, which depends on the proportion of the basin area covered by vegetation with deep roots.
Sm (mm)	Number		Areal mean free water capacity of the surface soil layer, which represents the maximum possible deficit of free water storage.
Ex	Number		Exponent of the free water capacity curve influencing the development of the saturated area.
Kg	Number		Outflow coefficients of the free water storage to groundwater relationships.
Ki	Number		Outflow coefficients of the free water storage to interflow relationships.
Cg	Number		Recession constants of the groundwater storage.
Ci	Number		Recession constants of the interflow storage.
Cs	Number		Recession constant in the lag and route method for routing through the channel system within each sub-basin
Ke	Number		Parameter of the Muskingum method
Xe	Number		Parameter of the Muskingum method
L	Number		Lag in time

Table 4. Descriptions of methods in *Precipitation_Forecast* interface

Method Name	**Description**
fp()	Forecasting precipitation by using certain hydrometeorological model with horizontal gridded data.
getResult()	Getting the result of forecasted precipitation sequence.

Table 5. Exceptions and their descriptions

Exception Name	**Description**
SystemException	A System Exception indicates that exceptions such as Database or network crashes.
UnAuthorizedException	An UnAuthorizedException is thrown if the user is unauthorized.
ASMException	An ASMException is thrown if there is no ASM data.
PeriodRainException	A PeriodRainException is thrown if there is no period precipitation data.
EvException	An EvException is thrown if there is no evaporation data.
DayRainException	A DayRainException is thrown if there is no daily precipitation data.

Table 6. Details of rain gauge stations in the study area

Station	**Kind of station**	**Station name**	**Weighting**	**Area(km^2)**
01	Rainfall	Jiangcun	0.0915	745
02	Rainfall	Daoxian	0.0846	691
03	Rainfall	Haofu	0.0689	562
04	Rainfall	Jiangyong	0.1134	926
05	Rainfall	Dalupu	0.1200	979
06	Rainfall	Centianhe	0.0326	266
07	Rainfall	Simaqiao	0.0651	531
08	Rainfall	Youxiang	0.0762	622
09	Rainfall	Ningyuan	0.0911	744
10	Rainfall	Shuishi	0.0651	532
11	Rainfall	Baijiaping	0.1210	988
12	Rainfall/streamflow	Shuangpai	0.0705	576

Table 7. Calibrated parameters of Xinanjiang model in case study

Parameter	B	I_m	K	C	U_m	L_m	D_m
Value	0.59	0.02	0.5	0.19	10.34	60.09	31.24

Parameter	S_m	E_x	K_g	K_i	C_i	C_g	C_s	K_e	X_e
Value	10.01	1.05	0.26	0.36	0.13	0.99	0.42	1.68	0.10

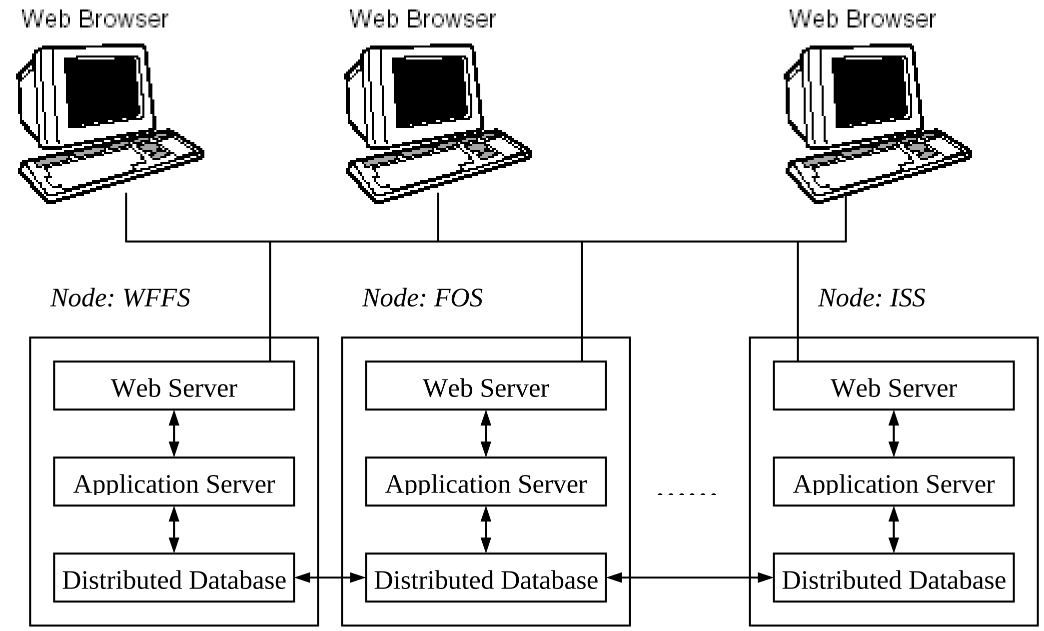


Fig. 1 Architecture of the Web-based distributed system

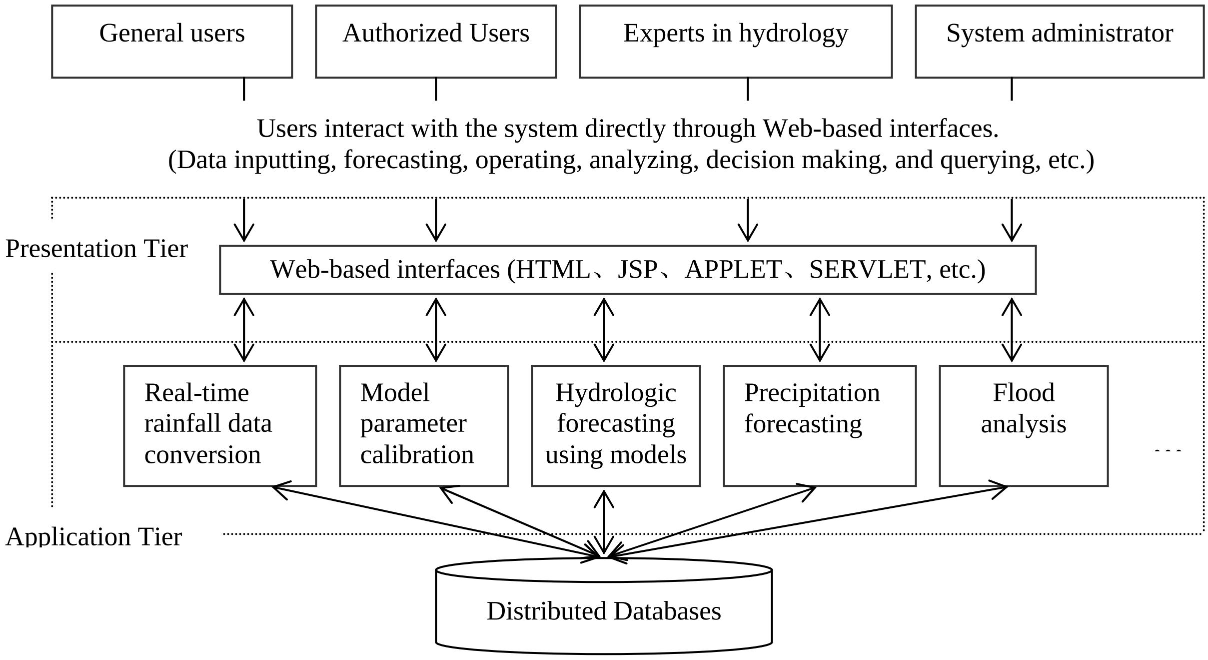


Fig. 2 Conceptual framework of WFFS

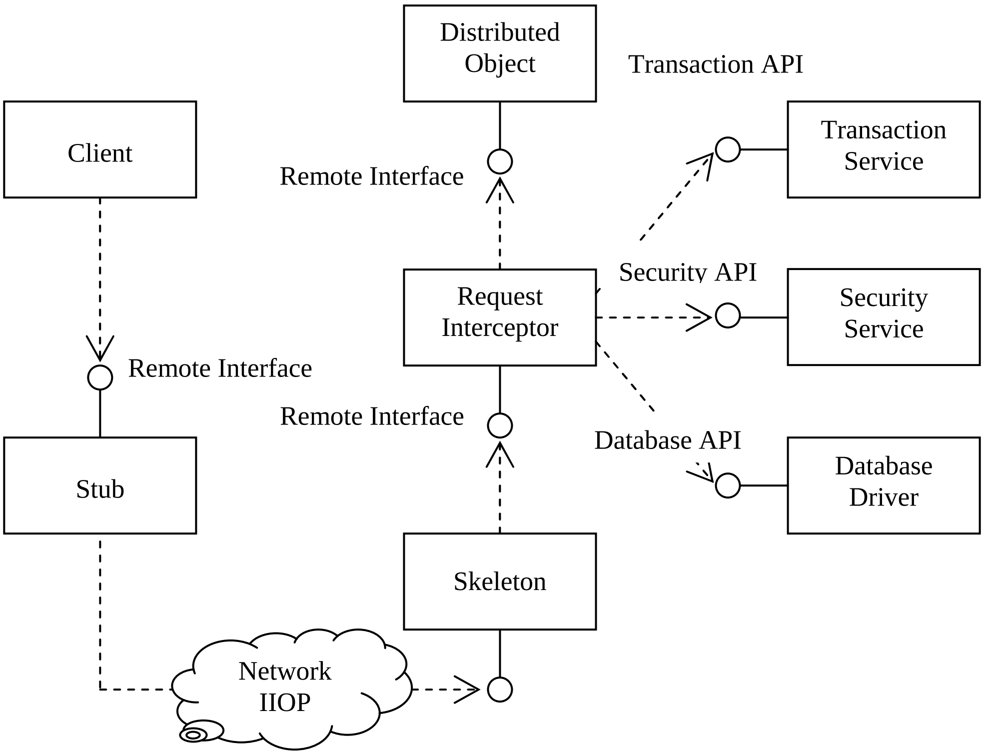


Fig. 3 Java Remote Method Invocation over IIOP

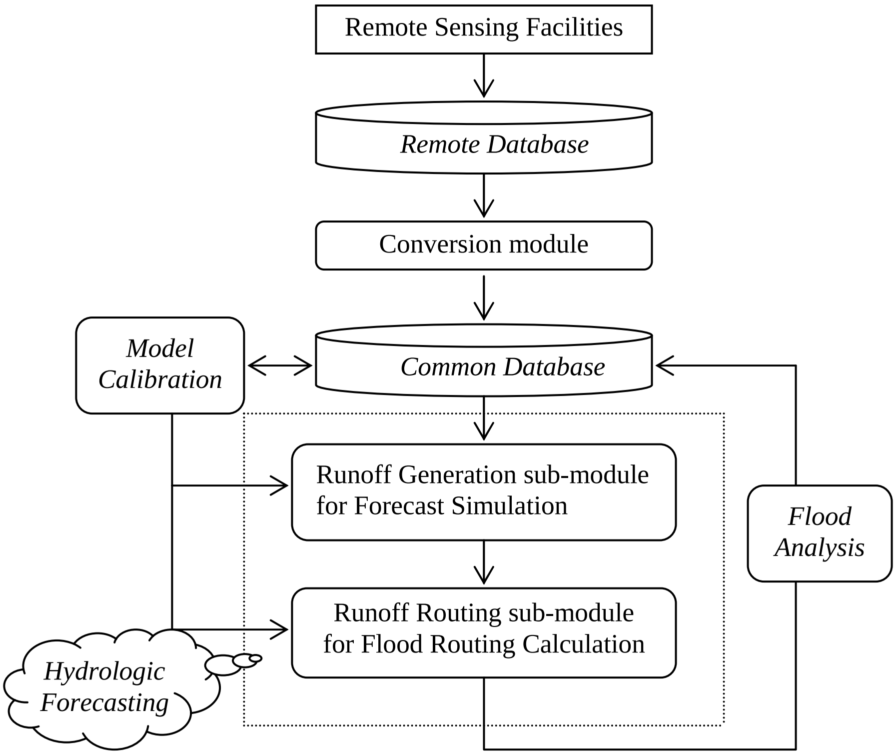


Fig. 4 Schematic diagram of forecasting process

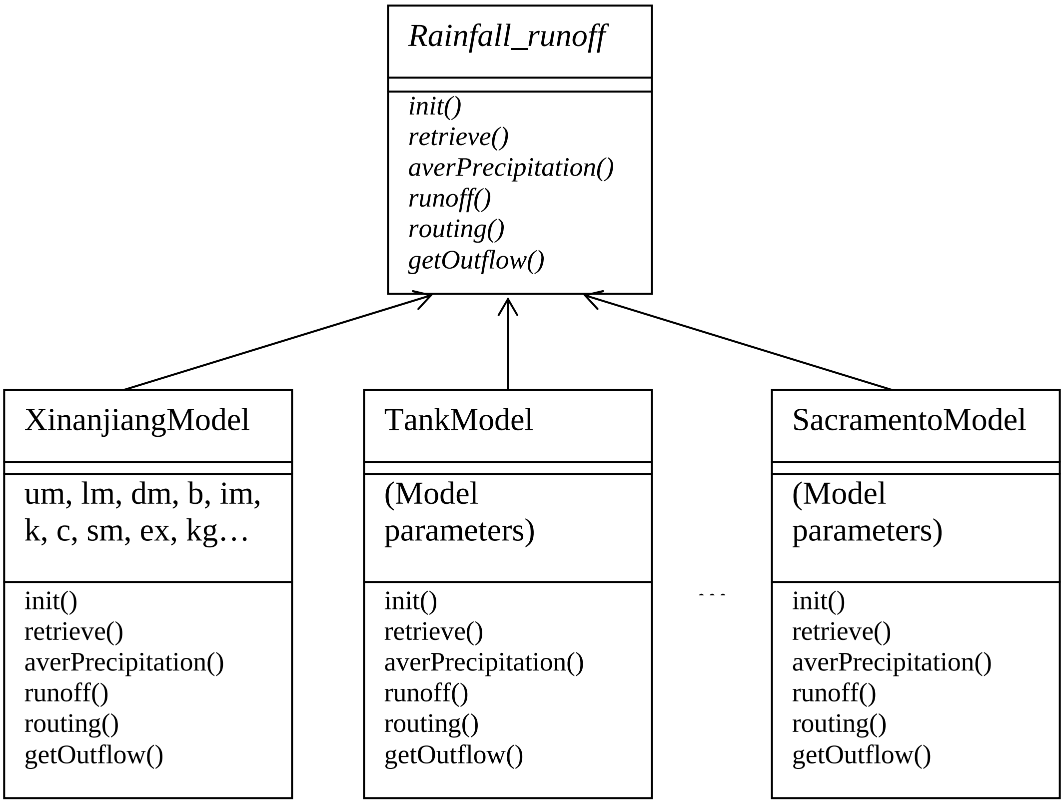
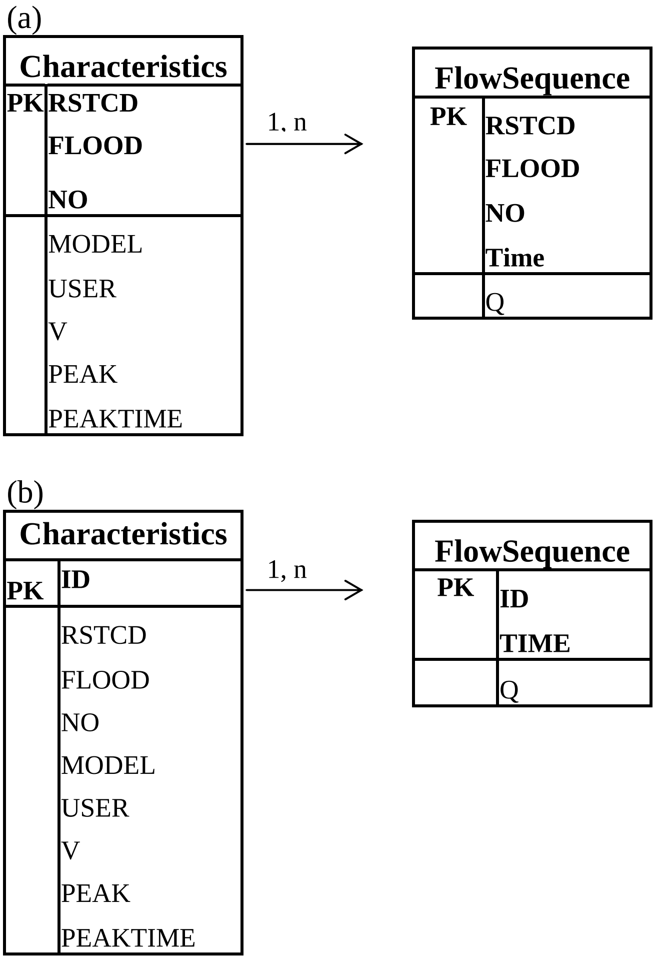


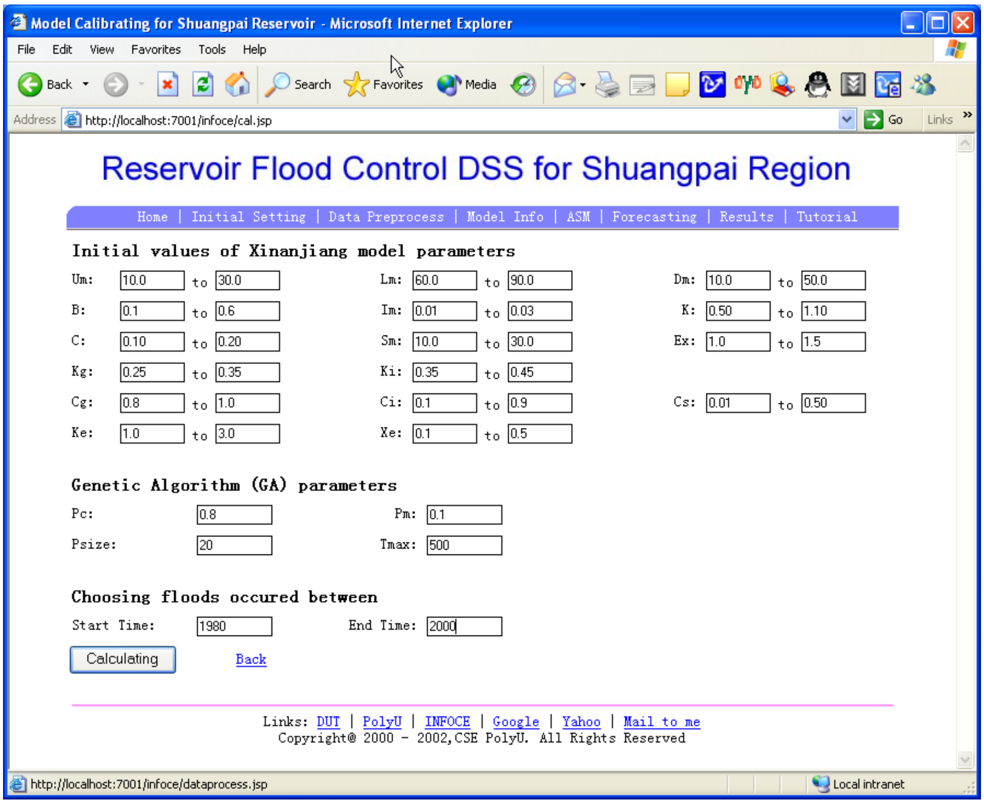
Fig. 5 Relationship between the interface and *Rainfall-Runoff Model* components



(c)

Field Name	Type	Description
ID	INTEGER	Identifier of an alternative that has no meaning.
RSTCD	CHAR	The standard code of the reservoir or river.
FLOOD	CHAR	Standard code of a flood.
USER	CHAR	User who made the alternative.
MODEL	CHAR	Hydrologic model applied in the alternative.
V	Number	Total runoff volume.
PEAK	Number	Peak discharge of the flood.
PEAKTIME	Number	Peak time of the flood.
NO	INTEGER	Serial number of an alternative.
Q	Number	Discharge of the main outlet.

Fig. 6 Definition of the table structure of flood characteristics



Reservoir Flood Control DSS for Shuangpai Region

Initial values of Xinanjiang model parameters

Um:	10.0	to	30.0	Lm:	60.0	to	90.0	Dm:	10.0	to	50.0
B:	0.1	to	0.6	Im:	0.01	to	0.03	K:	0.50	to	1.10
C:	0.10	to	0.20	Sm:	10.0	to	30.0	Ex:	1.0	to	1.5
Kg:	0.25	to	0.35	Ki:	0.35	to	0.45				
Cg:	0.8	to	1.0	Ci:	0.1	to	0.9	Cs:	0.01	to	0.50
Ke:	1.0	to	3.0	Xe:	0.1	to	0.5				

Genetic Algorithm (GA) parameters

Pc:	0.8	Pm:	0.1
Psize:	20	Tmax:	500

Choosing floods occured between

Start Time: 1980 End Time: 2000

Links: DUT | PolyU | INFOCE | Google | Yahoo | Mail to me
Copyright@ 2000 - 2002,CSE PolyU. All Rights Reserved

Fig. 7 Parameter input interface for model calibration

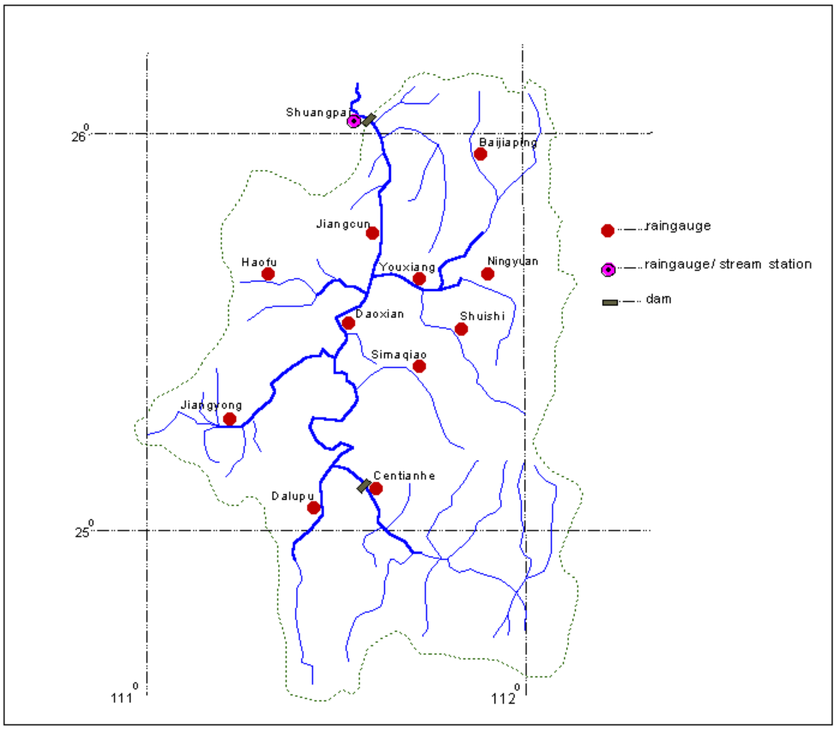
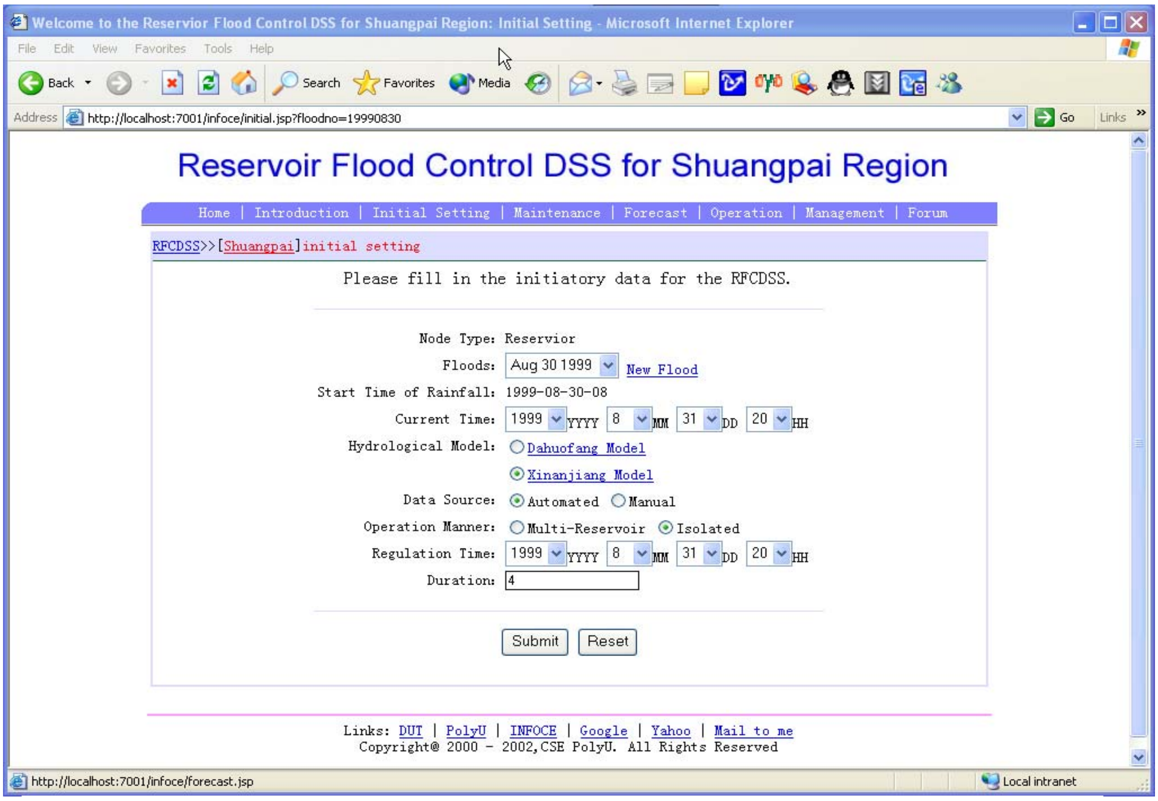


Fig. 8 Map of the Shuangpai area with locations of rain gauge stations



http://localhost:7001/infoce/initial.jsp?floodno=19990830

Reservoir Flood Control DSS for Shuangpai Region

RFCDSS>>[Shuangpai]initial setting

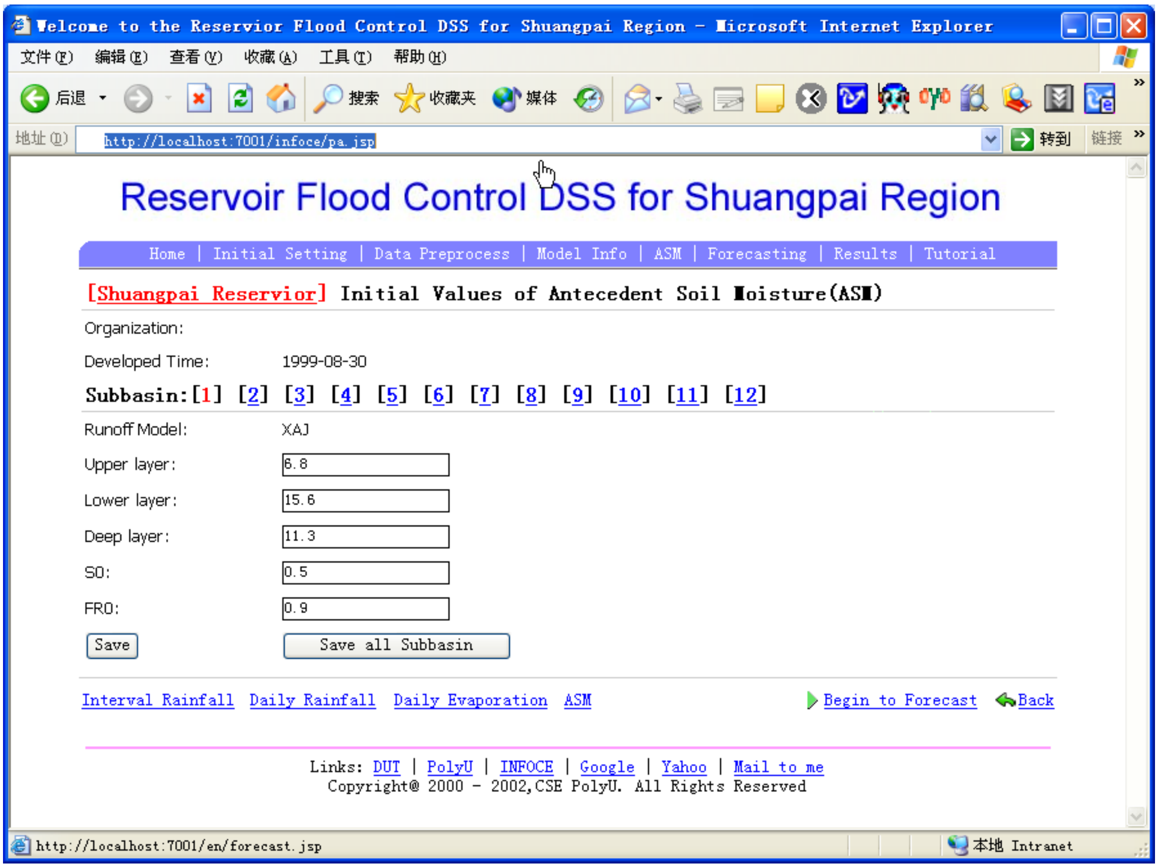
Please fill in the initiatory data for the RFCDSS.

Node Type: Reservior
Floods: Aug 30 1999 New Flood
Start Time of Rainfall: 1999-08-30-08
Current Time: 1999 YYYY 8 MM 31 DD 20 HH
Hydrological Model: ○ Dahuofang Model ● Xinanjiang Model
Data Source: ● Automated ○ Manual
Operation Manner: ○ Multi-Reservoir ● Isolated
Regulation Time: 1999 YYYY 8 MM 31 DD 20 HH
Duration: 4

Links: DUT | PolyU | INFOCE | Google | Yahoo | Mail to me
Copyright@ 2000 - 2002,CSE PolyU. All Rights Reserved

http://localhost:7001/infoce/forecast.jsp

Fig. 9 Initial setting for the RFCDSS



http://localhost:7001/infoce/pa.jsp

Reservoir Flood Control DSS for Shuangpai Region

[Shuangpai Reservior] Initial Values of Antecedent Soil Moisture(ASM)

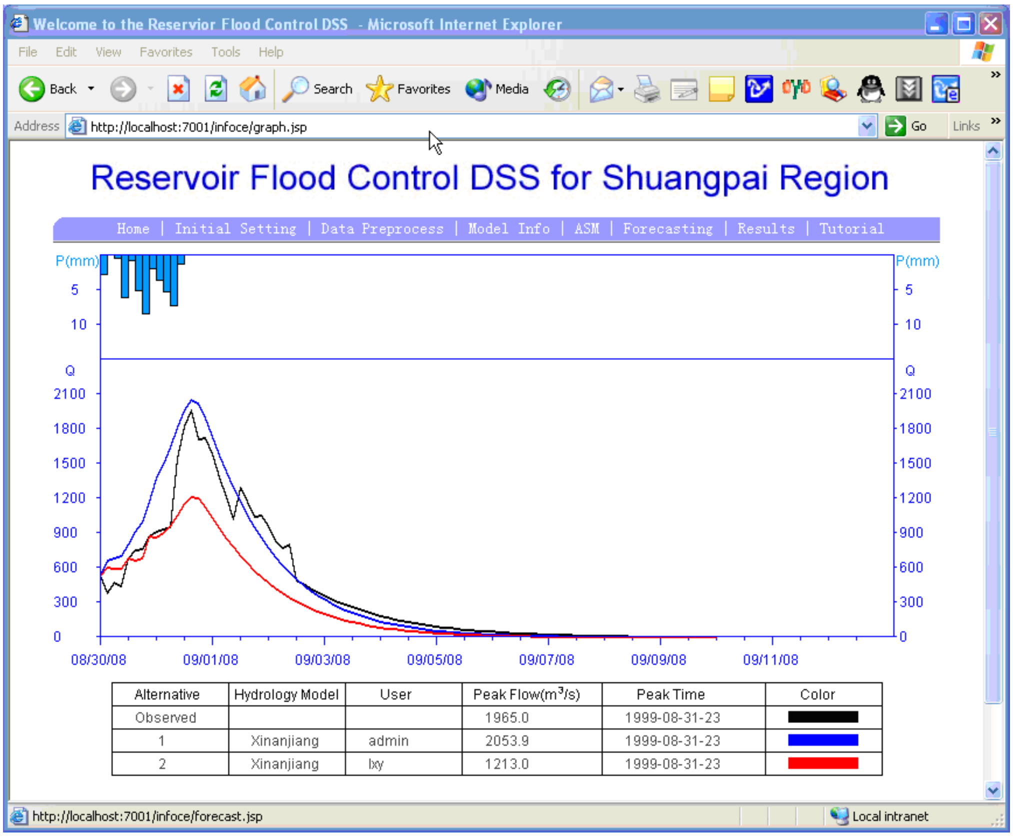
Organization:
Developed Time: 1999-08-30
Subbasin:[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]
Runoff Model: XAJ
Upper layer: 6.8
Lower layer: 15.6
Deep layer: 11.3
S0: 0.5
FR0: 0.9

Interval Rainfall Daily Rainfall Daily Evaporation ASM Begin to Forecast Back

Links: DUT | PolyU | INFOCE | Google | Yahoo | Mail to me
Copyright@ 2000 - 2002,CSE PolyU. All Rights Reserved

http://localhost:7001/en/forecast.jsp

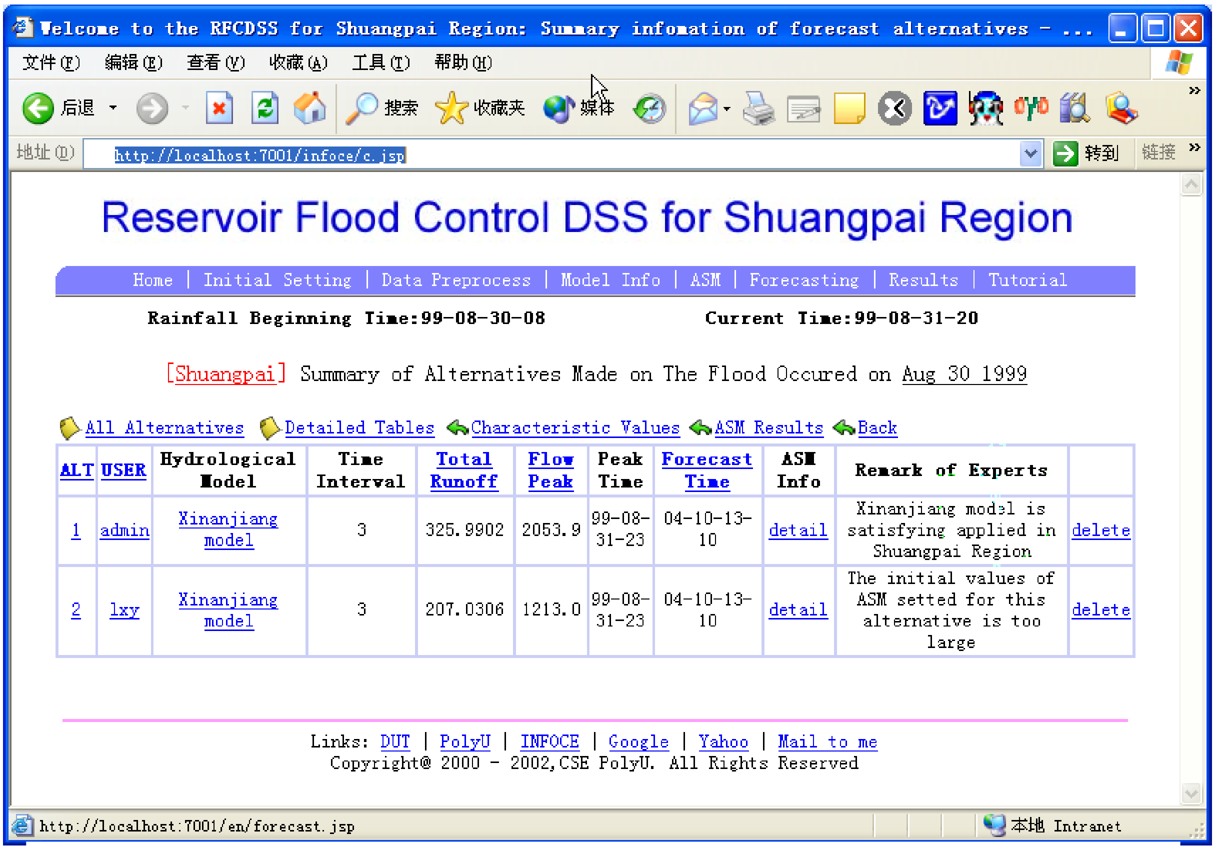
Fig. 10 Inputting web page of initial values of ASM



Reservoir Flood Control DSS for Shuangpai Region

Alternative	Hydrology Model	User	Peak Flow(m^3/s)	Peak Time	Color
Observed			1965.0	1999-08-31-23	■
1	Xinanjiang	admin	2053.9	1999-08-31-23	■
2	Xinanjiang	lxy	1213.0	1999-08-31-23	■

Fig. 11 Comparison of observed flow to forecasted flow



Reservoir Flood Control DSS for Shuangpai Region

Rainfall Beginning Time:99-08-30-08 **Current Time:99-08-31-20**

[Shuangpai] Summary of Alternatives Made on The Flood Occured on Aug 30 1999

ALT	USER	Hydrological Model	Time Interval	Total Runoff	Flow Peak	Peak Time	Forecast Time	ASM Info	Remark of Experts	
1	admin	Xinanjiang model	3	325.9902	2053.9	99-08-31-23	04-10-13-10	detail	Xinanjiang model is satisfying applied in Shuangpai Region	delete
2	lxy	Xinanjiang model	3	207.0306	1213.0	99-08-31-23	04-10-13-10	detail	The initial values of ASM setted for this alternative is too large	delete

Links: DUT | PolyU | INFOCE | Google | Yahoo | Mail to me
Copyright@ 2000 - 2002,CSE PolyU. All Rights Reserved

Fig. 12 Summary of all alternatives and experts' remark