

# Object-oriented Hydrological Modeling System

Y. Ichikawa, Y. Tachikawa, K. Takara

*Disaster Prevention Research Institute, Kyoto University, Uji 611-0011, Japan*

M. Shiiba

*Department of Civil Engineering Systems, Kyoto University, Kyoto 606-8501, Japan*

**ABSTRACT:** A new hydrological modeling system is developed on the basis of the object-oriented analysis of hydrological systems. In this modeling system, hydrological systems are considered as a set of dynamic elements which communicate one another. This modeling system provides the base element model which has the common functions required in hydrological simulations, such as initialization of state variables, giving numerical values to parameter variables, setting computation time steps. Also this system provides the port models for data transfer between element models. The system user can easily create his own element models by using the base element model and easily build the total system model for the hydrological simulations by connecting a number of element models through the port models.

## 1 INTRODUCTION

Hydrological systems are generally considered as a set of unit processes such as precipitation, evapotranspiration, water flow on the slope systems, river network flow, water storage in the dam reservoirs. Therefore hydrological simulation models are usually built using the following method in which:

1. the element models for each processes are prepared, and
2. the total system model for the hydrological simulations is built by connecting the element models according to the basin structure.

This method has the advantage that the total system model can be easily built and modified (Figure 1).

In order to support this modeling method, we developed the computer

software system, termed Object-oriented Hydrological Modeling System (OHyMoS). OHyMoS is designed on the basis of the object-oriented programming concepts and it enables the user to easily develop his own element models and easily build the total system model for the hydrological simulations by connecting a number of element models.

## 2 EXISTING MODELING SYSTEMS

There are some hydrological modeling systems which contain the libraries of element models. The users can select the element models from the libraries according to their needs. HEC-HMS, developed by US Army Corps of Engineers Hydrologic Engineering Center (1998), is such kind of modeling system. This system provides seven types of element:

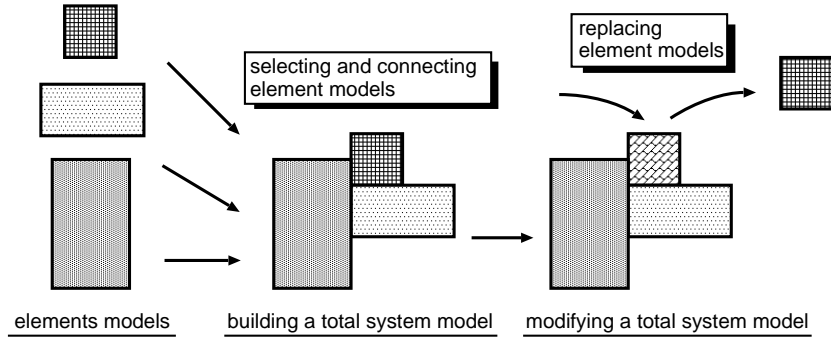


Figure 1: Building and modifying a total system model

subbasin, routing reach, junction, reservoir, source, sink and diversion. The user can construct a basin hydrological model by selecting and connecting these elements. Also each element has several options for hydrological computation, and the user can choose an appropriate option for each element. This type of modeling system has the advantage that the user is free from the laborious works of writing the codes of the element models if he uses the element models provided in the modeling system.

However, let us suppose that a user attempts to originally develop a new element model and connect it with other models provided in the modeling system. What should he know for his purpose? First, he has to know the structure of the element model; i.e. he has to know what variables and what functions to be required. Second, he has to know the method of data transfer between element models. If he does not know it, his element model could not be connected to other models. Furthermore, he may have to understand the specifications of the graphical user interface used in the modeling system. It means that he should have the detailed knowledge of the mod-

eling system, which is usually a huge and complex software. It seems to be a very strict requirement and it is practically impossible for the user to develop his own element model and to use his model with the provided ones. From similar reason, it seems to be difficult to modify the provided models.

### 3 BASIC REQUIREMENTS OF HYDROLOGICAL MODELING SYSTEM

By summarizing the previous section, it can be said that the existing modeling systems are not flexible enough to be customized according to the user needs, and a hydrological modeling system should be designed so that the user can easily develop his own element model and connect it to other element models. We propose that the hydrological modeling system must satisfy the four following basic requirements:

- (a) All of the element models must have a common structure.
- (b) The method of data transfer between element models must be standardized.

- (c) The element model structure and the data transfer method must be open to the users.
- (d) It must be easy for the user to develop his own element model according to the element model structure and the data transfer method.

Requirements (a) and (b) are the minimum requirements of hydrological modeling systems. Requirement (c) can be satisfied if a detailed documentation for the modeling system is provided. However, to satisfy requirement (d) is practically very difficult. Most existing hydrological modeling systems have been coded in the procedure-oriented computer languages, such as C or Fortran, and have generally prepared the function libraries for data transfer, model operation, graphical user interface, etc. Even if the detailed documentations for the function libraries are provided to the system user, he must read the documentations and understand what functions to be required and how to use them in order to write the codes of his own element model. This is a hard task and makes it difficult for the user to develop element models. To satisfy requirement (d) is the key to development of a flexible modeling system.

## 4 OBJECT – ORIENTED DESIGN FOR HYDROLOGICAL MODELING SYSTEM

### 4.1 Basic concepts

In order to satisfy the basic requirements proposed in the previous section, especially requirement (d), we adopt the object-oriented approach to develop a hydrological modeling system. In this approach,

- hydrological elements are regarded as “objects” and basin hydrological systems as a hierarchy of objects of the hydrological elements,
- the functions for an element model are divided into peculiar functions and common functions; e.g. initializing the state variables, giving numerical values to the model parameters, setting computation time step (Figure 2).
- the common functions are packed into “*the base element model*” (Figure 2), and
- the user can develop his own element model by combining its peculiar functions with the base element model (Figure 3).

The most important point in this approach is the base element model. A model developer can use the base element model as the “template” for his model and add appropriate functions to his model as he likes. Because his model will automatically inherit all the common functions defined in the base element models, he does not have to write the codes for these functions and can devote himself to the part of hydrological computations. Moreover, all the element models will have the common structure because they are all derived from the base element model. This fact implies that all the element model can be operated through the same function calls.

### 4.2 Methodology

We code the hydrological modeling system in the object-oriented computer language, C++, which has the three object-oriented programming concepts: 1. class, 2. inheritance, and 3. polymorphism.

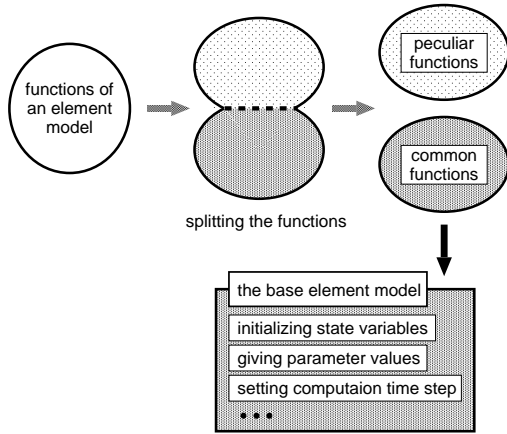


Figure 2: Base element model

#### 4.2.1 Class

A *class* is a package that contains several data variables and several functions to operate data variables. The data variables are called *data members* and the functions *member functions*.

Because an element model executes hydrological computations by operating state variables and parameters in the hydrological equations, it can be expressed as a class by regarding the state variables and the parameters as the data members and the equations as the member functions.

#### 4.2.2 Inheritance

By using *inheritance*, new classes can be created on the basis of existing classes. For example, suppose that we want to create a new class which will have the data members and the member functions defined in an existing class with further data members and member functions (Figure 4). In this case, by deriving the new class from the existing one, we do not have to write the common part of these classes in the implementation of the new class. The new class will auto-

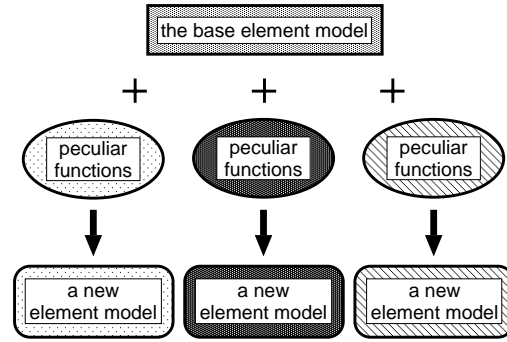


Figure 3: Developing a new element model

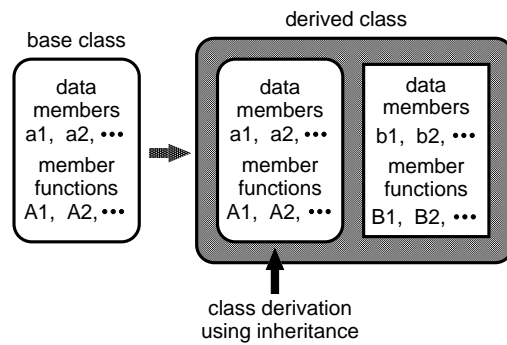


Figure 4: Inheritance

matically inherit the data members and the member functions defined in the existing class. Of course, we can add further data members and member functions to the new class. The new class is called a *derived class* and the existing one a *base class*.

In the modeling system, we provide the base element model as a base class and then an user will be able to derive his own element model from the base element model by using inheritance. His element model will automatically have the common functions provided in the base element model.

### 4.2.3 Polymorphism

Inheritance is used when a new class will have the same member functions as an existing class. However they may have the same member functions with the different implementations, i.e. the member functions whose purposes are same but whose methods are different. C++ allows this situation by using *polymorphism*. Polymorphism enables us to implement the inherited member functions differently from those of the existing class.

For example, all the element models must have the member function to calculate hydrological phenomena. It is therefore quite natural that this member function should be defined in the base element model and all the element model should have this function by using inheritance from the base element model. However it is evident that each element models must implement this function uniquely. To solve this dilemma, we use polymorphism. By using polymorphism, the element models can have each variants of the function inherited from the base element model.

## 5 OBJECT – ORIENTED HYDROLOGICAL MODELING SYSTEM

### 5.1 Element model, subsystem model, total System model

In the previous section we described the concept of an element model. We should explain the concept of a *subsystem model* and a *total system model* too. A subsystem model is a set of element models and a total system model is a set of subsystem models and element models. Suppose, for example, the study basin consists of three subbasins and each subbasin consists of a slope area and a

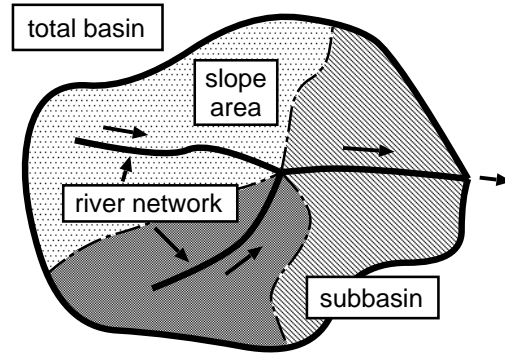


Figure 5: Basin structure

river network (Figure 5). In OHyMoS, first, the slope area and the river network are modeled as the element models. These element models are derived from the base element model. Second, the subbasins are modeled as subsystem models by connecting the slope area element model and the river network element model. Finally, the study basin is modeled as the total system model by connecting three subsystem models (Figure 6). Note that the user must build the total system model for his study basin for himself. How to build a total system model will be given in 5.4. Unlike a total system model the user dose not need to use the subsystem models; i.e. the model structure shown in Figure 7 is possible.

An element model does actual hydrological computations according to the hydrological characteristics and a subsystem model and a total system model work as managers of the element models and/or the subsystem models; they issue several commands to operate the subordinate models. Also, a total system model works for communicating with a user interactively through command line and dealing with data files.

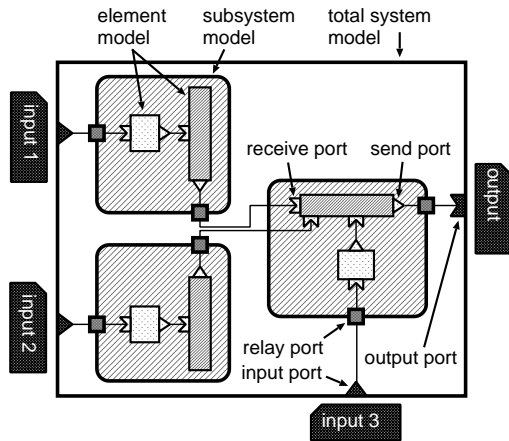


Figure 6: Model structure using OHyMoS

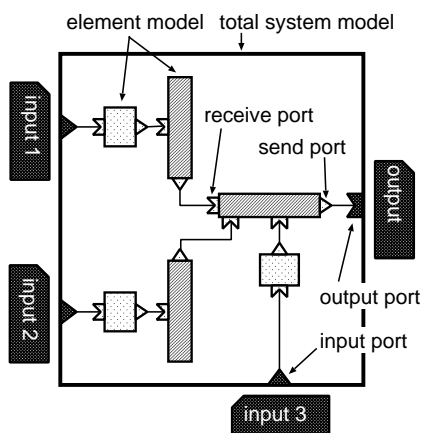


Figure 7: Another model structure

## 5.2 Data transfer

OHyMoS also provides the model for communicating (sending and receiving data) between the element models. Using this model, a model developer does not need to write the codes to transfer the data from his element model to other element models. He has only to add this model to his own element model and write the codes to define the relationships between the element models.

The model for sending data from an element model is called a *send port* and the model for receiving data from another element model is called a *receive port*. A subsystem model has *relay ports* for data transfer with the outside. A total system model has *input ports* to read data files (rainfall, temperature, etc) and *output ports* to write the calculation results (discharge, water depth, etc) to files. The data transferred through the ports is modeled as a *data pack*, which contains the time and the numerical value(s).

The port models and the data pack model also have the object-oriented hierarchies whose roots are *the base port model* and *the base data pack model*, respectively.

## 5.3 How to develop an element model

It is the strongest point of OHyMoS that the user can easily develop his own element model. The procedure for developing an element model is as follows:

1. derive an element model from the base element model,
2. define parameter variables and state variables as the data members,
3. define send/receive ports as the data members,
4. implement the member functions which register send/receive ports,
5. implement the member functions which give initial values to the state variables and give numerical values to the parameter variables,
6. implement the member function which sends data at the beginning of the simulation,
7. implement the member function which determines the time step for the next computation,
8. implement the member function which judges whether or not to exe-

- cute the next computation,
9. implement the member function which does hydrological computations and sends the result data, and
  10. implement the member function which drives a series of computational work until the simulation ends.

5.4 How to build a total system model  
OHyMoS provides *the base total system model* (and *the base subsystem model*) so that the user can easily build the total system model for his study basin. The procedure for building the total system model is as follows:

1. derive an total system model from the base total system model,
2. define the subordinate element / subsystem models as the data members,
3. implement the member function which registers the subordinate models,
4. define the input/output ports as the data members,
5. implement the member functions which register the input/output ports,
6. implement the member function which connects data files to the input/output ports, and
7. implement the member function which connects the input/output ports and the send/receive ports between the subordinate models.

5.5 Main program for a simulation

The main program for a simulation consists of a set of commands to operate the total system model. When receiving the commands from the main program, the total system model will issue the messages to drive the subordinate models.

## 6 APPLICATION

OHyMoS was applied to Huaihe River basin in China (145,000 km<sup>2</sup>) in order to simulate rainfall-runoff processes. Figure 8 shows Huaihe River basin. This basin was divided into 1478 subbasins by 10' × 10' grid boxes. Each subbasin was modeled as a subsystem model which consisted of the xinanjiang model (Zhao et al., 1980) representing runoff processes from the slope area and the lumped kinematic stream network model (Shiiba et al., 1996) representing flood wave in the river network (Figure 9). The whole basin was modeled by connecting the subsystem models (Figure 10). For detail of these models, you can also refer to Tachikawa et al.(2000). Figure 11 shows the simulation result.

## 7 CONCLUSION

A hydrological modeling system has been presented on the basis of the object-oriented approach and design. In this modeling system, hydrological elements are regarded as objects and basin hydrological systems as a hierarchy of objects of hydrological elements. This modeling system provides the base element model from which the user can derive his own element model and the port models for data transfer between element models. The user can easily create his own element models by using the base element model and easily build the total system model by connecting a number of element models through the port models.

We distribute this modeling system at <http://rdp.dpri.kyoto-u.ac.jp/lab/ohyMos.html> and <ftp://rdp.dpri.kyoto-u.ac.jp/pub/>. If you use our system and have some comments on this system, please let us know.

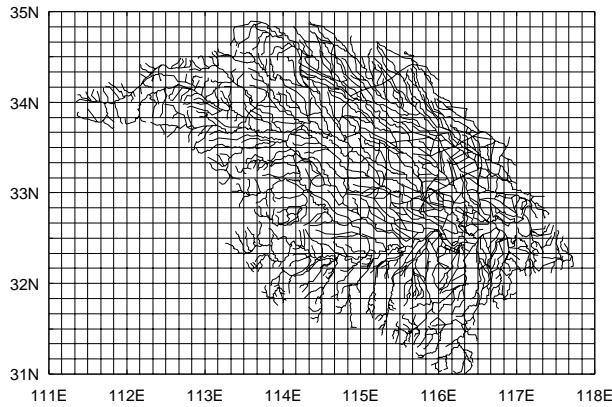


Figure 8: Huaihe River basin

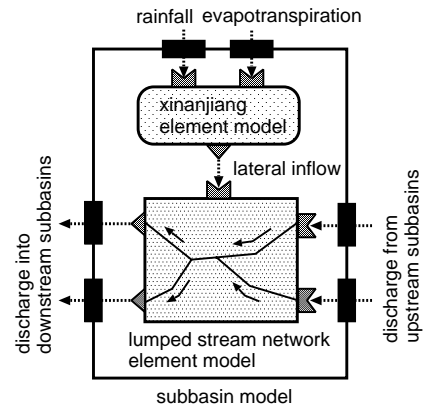


Figure 9: Element models and sub-system model

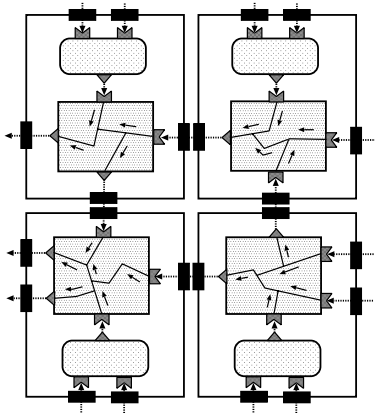


Figure 10: Total system model

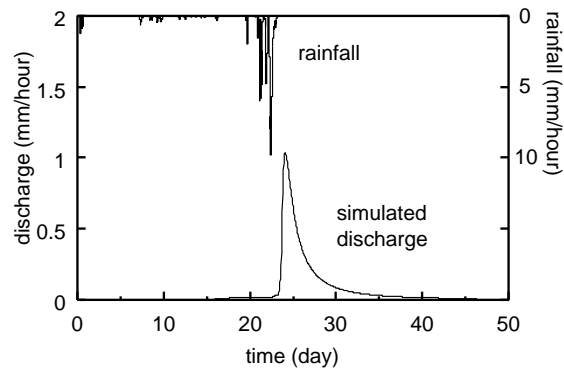


Figure 11: Simulation result

## REFERENCES

- US Army Corps of Engineers Hydrologic Engineering Center, "HEC-HMS Hydrologic Modeling System User's Manual," version 1.0, 1998.
- Zhao, R.J., Y.L. Zhang, L.R. Fang, X.R. Liu and Q.S. Zhang, "The Xinanjiang model," Proc. of the Oxford Symposium on Hydrological Forecasting, IAHS Publ. no. 129., pp. 351 – 356, 1980.
- Michiharu Shiiba, Yutaka Ichikawa, Shuichi Ikebuchi, Yasuto Tachikawa

and Takuma Takasao, "Scale-up of a runoff model using GIS and an object-oriented hydrological modeling system," HydroGIS 96, IAHS Publ. no. 235, pp. 63 – 70, 1996.

- Yasuto Tachikawa, Yutaka Ichikawa, Kaoru Takara and Michiharu Shiiba, "Development of a macro scale distributed hydrological model using an object-oriented hydrological modeling system," Submitted to Hydroinformatics 2000, 2000.