PINE – A Flexible Hydrological Modeling System

<u>Trond Rinde</u>, SINTEF – Civil and Environmental Engineering, Department of Water Resources, Klaebuveien 153, N-7465 Trondheim, Norway (trond.rinde@civil.sintef.no)

Abstract: A software system called Process Integrating Network (PINE) is presented. The system is a simulation tool for hydrological and hydrologically related processes. It is based on object-oriented principles and has been particularly designed for freedom in choice of model structures and algorithms for process description. Simulation models are built by linking process components together in a structure. The system can handle compound working media, such as water with chemical or biological constituents. Non-hydrological routines may then be included to describe the responses of such constituents. Geographical information systems may be integrated to obtain spatial distribution in system parameterisation, process simulation, and visualisation of simulation results. Extensibility and reuse of program components have been emphasised in the program design. Separation between process topology, process descriptions, and process data on the code level, allows for simple and consistent implementation of process modules. Such modules may be automatically prototyped, and implementation of their response functions can be done without knowledge of the rest of the system, and without the need to implement import or export routines, or user interface. Model extension is thus rapid and does not require extensive programming skills. The components for process description may be placed in program libraries, which can be included in the program when required. The program system can thus be very compact, while it still has a large number of process algorithms available. The system can run on both PC and UNIX platforms.

1. INTRODUCTION

Over the years hydrological simulation systems have migrated from simple, conceptual models towards physically based and spatially distributed models. Most systems, however, are still static with regard to model structure and methods of process description. Special adaptation to unusual simulation situations is therefore difficult. Model development has also typically been directed towards research applications. Models for operational use have therefore to a lesser extent moved towards higher levels of sophistication, see for instance Blöschl and Sivapalan [1995] and Abbott and Refsgaard [1996]. This is partly a result of the large costs associated with taking the new and detailed models into use. However, few alternatives exist that facilitate a gradual increase in modelling detail. The PINE system, presented in this paper, offers a flexibility that allows both adaptations to unusual simulation situations, and gradual transitions towards higher levels of simulation detail. It also allows hydrological modellers to build, or restructure simulation models without, or with minimal, need for programming.

1.1 The PINE concept

The PINE concept is based on the idea that hydrological systems can be represented as sets of interrelated system-elements. Such a set of interrelated system-elements is called a system-model (Figure 1b). System-elements may contain processes, which generate states and responses, and they may be related through exchange of mass, energy, or momentum. The combined behaviour of the system-elements in a system-model reflects the behaviour of the system that is modelled by it.

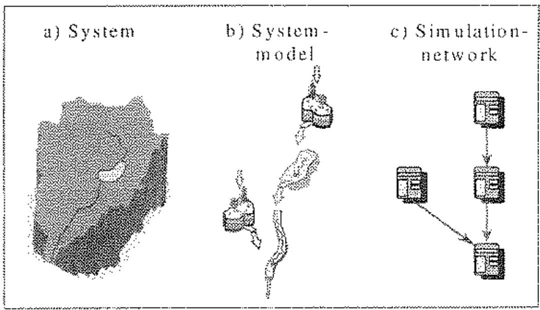

Figure 1 System-model and simulation network.

A system-model can in PINE be realised in terms of a simulation network built up of nodes and links (Figure 1c). The nodes, which are called system-nodes, then correspond to elements in the system-model and they contain data elements and process routines that enable them to express the properties and the behaviours of these elements. Different types of system-elements are hence represented by system-nodes with different sets of data elements and process routines. Links between system-nodes in a simulation network reflect relations between system-elements. The most typical relations between elements in hydrological system-models are paths for water flow.

1.2 Using PINE

With the PINE system, simulation-networks may be established at run-time and immediately executed. The program user simply needs to define the necessary number of system-nodes, assign appropriate process routines to them, and link them together according to how their corresponding system-elements are related. Finally he or she must specify the data elements that shall be supplied as input, and those that shall be exported as output. When that has been done, simulations can be run by simply supplying the input data and initiating the simulation loop.

As long as adequate process algorithms exist in the system, program users may map system-models specified on a sheet paper into executable simulation programs in just a matter of minutes. The users may also gradually increase the detail in their system representation, by introducing new system-nodes for additional system-elements they want to describe. In cases where appropriate process descriptions do not exist, new ones can be generated with only minimal programming effort.

2. THE WORKING PRINCIPLES OF PINE

The PINE program is made up of six main modules (Figure 2). At the code level these are highly separated in order to support easy replacement and individual development of program components. In particular, the components for process description are given a simple and consistent design. Implementing new such components is therefore easy and does not require considerations about import or export functionality or user interaction. The main modules are further made dynamic to facilitate flexible configuration of simulation setups at run-time. Three of the modules make up the system's simulation kernel, whereas the rest support and administer this kernel during program execution.

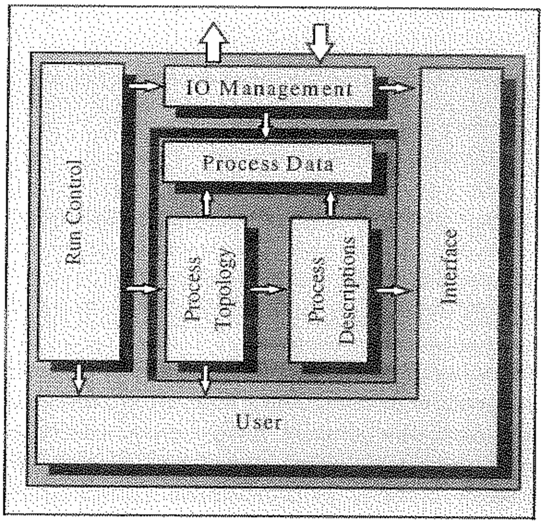

Figure 2 Main modules in PINE.

The modules for IO-management and user interaction are responsible for exchanging information to and from permanent storage and between the system and the program users. The run control module administers the simulation runs by requesting for the necessary user input, invoking the simulation algorithms, and trigger import and export of data. The process topology module, the process description module, and the process data module constitute the simulation kernel of the system. These modules co-operate closely during simulation runs. The process topology module stores system-nodes and links between them in computer memory. The process description module similarly stores process algorithms and makes these accessible to the system-nodes. The process data module finally maintains the data elements that hold values for the simulation variables, and makes these available to the other program modules. The program components involved in the construction of simulation-networks are further described below.

2.1 System-nodes

The system-nodes are the fundamental building blocks of PINE. A system-node must be defined for every system-element that is to be explicitly described by a simulation model. A system-node contains three principal elements. These are: (1) references to process description components, (2) a reference to a data container with simulation data elements, and (3) references to other system-nodes with which the system-node interacts (Figure 3).

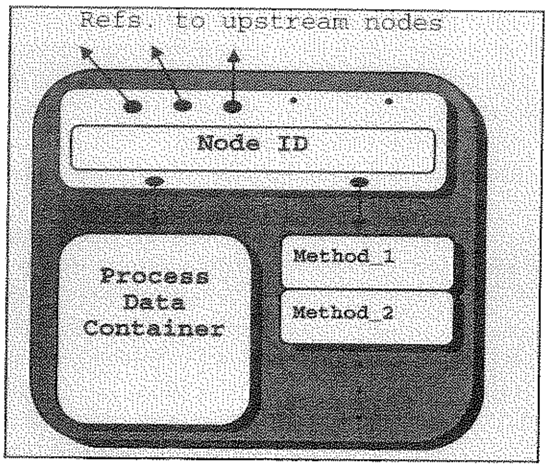

Figure 3 System-node.

A system-node is used to represent a body of water located at some place in the system that is represented. In order to maintain mass balance, every node must at least have data elements for inflow, storage and outflow of water, in their data containers. Depending on the numbers and types of simulation-methods that are attached to the nodes, the containers will usually have other data elements as well. Simulation-methods may share data elements in the data containers. During executions, the simulation-methods work on the data elements in the data containers. The data elements may later be accessed by other parts of the program, for export or display of simulation results.

System-nodes may receive inflow from any number of upstream nodes. The inflows may then be weighted relative to each other, or transformed into common units. In an execution, a node first collects its inflows from upstream suppliers, and assigns it to a variable in its data container. Then it invokes the simulation-methods, which generate values for the other data elements in the data container. Figure 4 illustrates how a system-element can be represented by a system-node with data elements and response routines. The method *CalcOutFlow* here calculates the outflow from the lake.

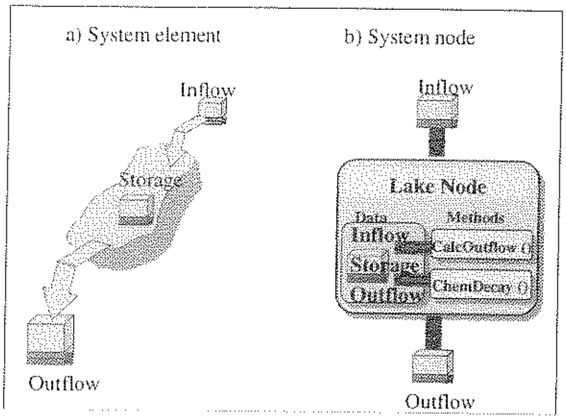

Figure 4 System-element and system-node.

2.2 Simulation-methods

The components for process descriptions, the simulation-methods, contain four main elements. These are (1) a name, (2) parameters, (3) variables, and (4) a process algorithm (Figure 5). In addition they have functionality for user interaction, and for storing and retrieving themselves to and from permanent storage.

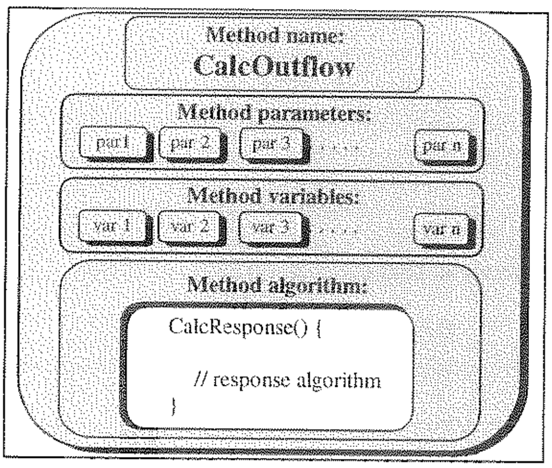

Figure 5 Simulation-method.

When a simulation-method is attached to a system-node, it inserts its variables into the node's data container. During execution, it accesses the elements there. Method parameters are, on the other hand, stored in the simulation-method itself. These data elements are thus specific to their "owner" and cannot be shared with other methods.

The method names are used to identify simulation-methods when configuring simulation setups.

2.3 Simulation data

The data container stores all the variables that are used by the simulation-methods within a system-node. In cases where more than one method makes use of the same variable, this principle prevents redundancy and enhances consistency among simulation data in the program. Method variables can reflect compound quantities. For instance, a method variable for inflow to a lake can contain another variable for the concentration of some chemical in the water. Separate simulation-methods can then be used to simulate the responses of such components. In Figure 4 for instance, the method *ChemDecay* calculates degradation of a chemical in the lake.

2.4 Simulation-networks

To construct a simulation model in PINE, program users must define a set of system-nodes, link them together into a structure, and assign simulation-methods to them. The types of simulation-methods used, defines the responses a model can produce. For program users, the program components that maintain process data operate behind the scenes. These components automatically become activated as soon as simulation-methods are adopted.

Figure 1 shows how a watershed may be mapped into a system-model, which in turn can be represented by a simulation network. This is, however, a very simplistic representation of a watershed, with entire sub-catchments represented by single system-nodes. A more detailed representation would be needed, i.e. to describe the responses of sub-regions in the catchment. Such an increased detail can easily be achieved by introducing system-nodes for area-regions, or even components within area regions, in the catchment. Such a detailed representation is shown in Figure 6, where separate system-nodes are used to model different snow-zones, area regions, and river sections.

2.5 Links between system-nodes

Links between system-nodes can be either flow links or control links. Flow links are used to exchange working medium between system-nodes, whereas control links are used to exchange control information such as state or parameter values.

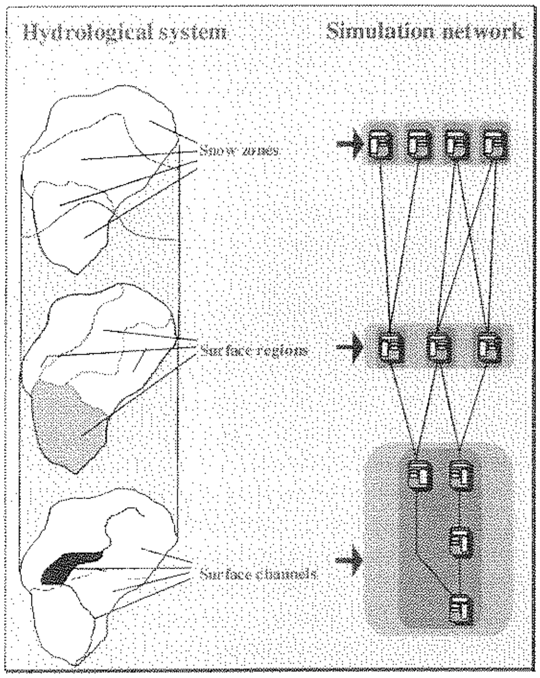

Figure 6 Detailed simulation-network.

A flow link is in fact a reference that connects a variable in one node to the inflow variable in another node. When establishing flow links, a user can freely choose which simulation variable in a node that is to be linked to the inflow variable in another. An inflow variable in a node may have flow links to several variables in one or more other nodes. Flow links are classified as either active or passive. Active flow links define the skeleton structure of simulation-networks. During simulations, the networks are traversed recursively along these links, and system-nodes are executed. Active flow links hence define the order in which system-nodes are executed. They can therefore not form circular flow relations in a network structure, since it would lead the traversal into an infinite loop. For such situations a passive flow link must be used at one place in the circular flow relation. These do not affect the execution order between system-nodes, but merely collect an existing response in the node they are linked to and deliver it to the node they are linked from.

Control links are used for exchange of state or parameter information between system-nodes. Control links allow process variables and parameters to reference data elements in other nodes. The variables or parameters will then always adopt the value of the data elements they refer to. The use of control links is normally required when there are either inter-node state-dependencies, or when process-parameters in many system-nodes always shall have identical values. The parameters may then be linked to a common "parent-parameter" in a monitor node. Editing the parent parameter will then automatically update the other parameters as well. Consistency is hence secured.

2.6 Batch processing

Under normal conditions, users interact with PINE through user dialogues. However, PINE may also be run without user interaction, from the command line. All information needed for a simulation run is then supplied from setup files whose names are specified as command-line options. The batch-processing option allows a more expedient simulation of complex systems, as well as facilitating smooth integration between PINE and other types of software programs.

To avoid extreme complexity in the simulation setups, it can be practical to break up very large or detailed hydrological systems into simpler sub-systems which can be modelled separately and whose responses can be combined to form the total system response. PINE supports such an approach through the batch-processing option. Complex systems may then be simulated by representing sub-systems in separate simulation-networks, and executing them in such a way that they yield input data to each other. This principle is illustrated in Figure 7.

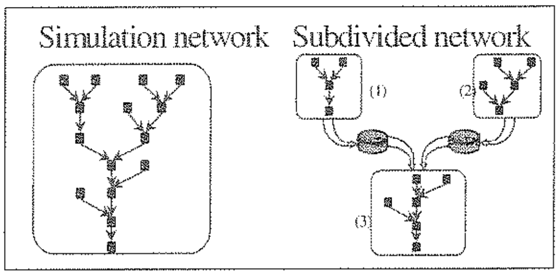

Figure 7 Modularised system simulation.

The batch-processing option also facilitates smooth integration between PINE and other program systems, since it allows the systems to run PINE as a child-process. This has for instance been utilised for integration of PINE with an integrated river-system simulation package, Killingtveit et al. [1998], and with a system for automatic model-calibration, Kolberg et al. [1998].

2.7 Integration of GIS

PINE supports fully distributed simulations through integration with a GIS. Distributed simulation-methods can be implemented as simply as lumped response algorithms. The methods then utilise GIS functionality to generate the distributed state and response values. The GIS further provides facilities for storage, manipulation and visualisation of distributed simulation data. It can therefore be used for preparation of simulation data, and for analysis and presentation of simulation results. Presently the GIS package GRASSLAND, LAS [1996] is used together with PINE, but any equivalent system could do.

In Figure 8 a distributed system representation is shown. Distributed simulation-methods are here used to describe snow, surface, and soil processes, and combined with lumped simulation-methods for transport through river channels and lakes.

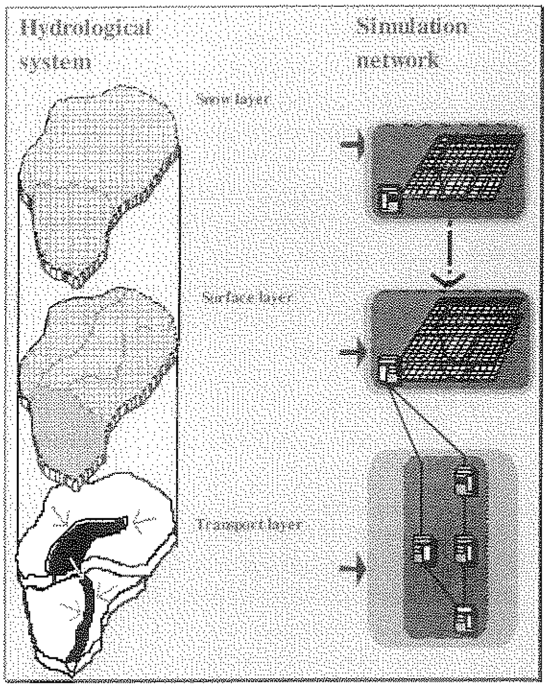

Figure 8 Distributed simulation system.

3. REPRESENTING THE HBV-MODEL IN PINE

The HBV-model, Bergström [1976], is a lumped, conceptual, hydrological model with components for snow accumulation and melting, soil-water retention, baseflow, and surface runoff. The model structure is shown in Figure 9. Representing the HBV-model in PINE only requires the algorithms for the model's components to be implemented as PINE simulation-methods. A simulation-network reflecting the HBV structure may then be established. Figure 10 shows a possible HBV-representation in PINE. Extensive flexibility is here introduced compared to the original model. Users can add or remove snow nodes, soil nodes, and nodes for precipitation or temperature stations, or choose to describe one or more processes in a distributed fashion. It is thus possible to customise the model to special modelling tasks or to non-natural features in the catchments. For instance, routines may be introduced for diversions, reservoirs and hydropower stations, in order to

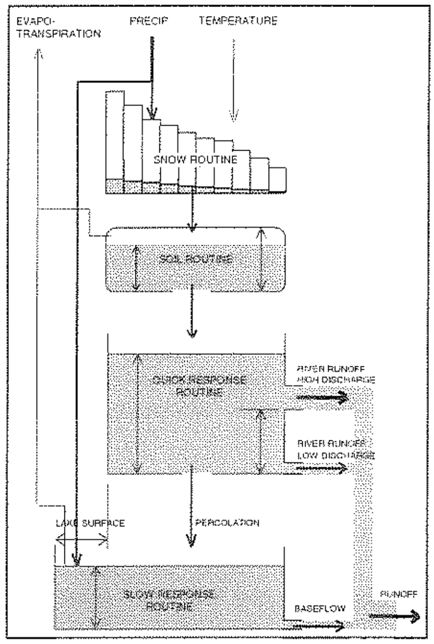

Figure 9 The HBV-model.

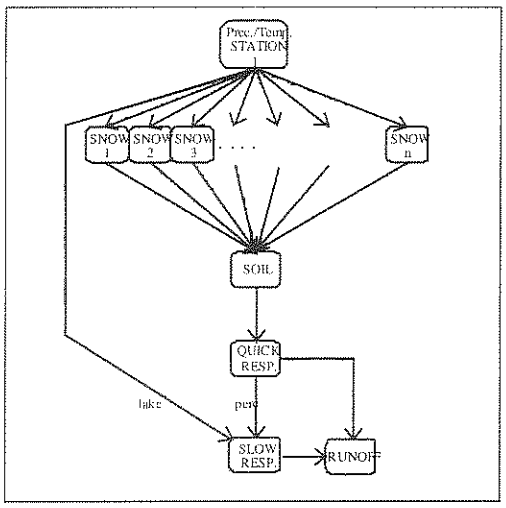

Figure 10 PINE representation of HBV.

allow simulation of regulated river systems.Distributed simulation-methods for snow processes may allow direct comparison between simulated snow covers and snow covers extracted from aerial photographs or satellite images. This may facilitate a more precise updating of snow storage in the model, and thus lead to improved spring-flood forecasts.

4. EXTENSIBILITY IN PINE

Three kinds of extensions have been prepared for in the program design. These are integration of new simulation-methods, adaptation to new standards for user interface, and adaptation to new systems for data storage.

Simulation-methods are defined by a name, a set of parameters and variables, and by a response algorithm. Program code for the data structures that hold these components are standardised and combined into an overall structure called a method class. The program can automatically generate such method classes. Implementation of a simulation-method therefore only requires a developer to define the parameters and the variables that will be used, and to insert the response algorithm into an automatically generated method class. The new simulation-method will then automatically integrate with the rest of the program. Data import and export, and user interaction is handled outside the method classes and does therefore not need to be considered.

The user interface module in PINE consists of user dialogue classes. When user-interaction is required, an object of a dialog-class is created. The object prompts for user input and returns the inserted information to the program component that created it. The dialog-object is subsequently destroyed. Switching user interface only requires implementation of a new set of dialogue classes, and possibly that the program's "main loop" be moved into the "main function" of the new user interface environment.

The IO management module in PINE contains a component called an import-export unit. This is the component that performs the actual transactions with the external storage system when data values are imported or exported. Switching from one storage system to another, e.g. by going from a file system to a database system, is therefore straightforward, since it only requires this unit to be replaced with an equivalent one that interacts with the new storage system.

5. CONCLUSIONS

The PINE program has been shown to offer a wide range of opportunities for the realisation of simulation models for different types of hydrological systems. Optimal solutions for simulation problems may be found via appropriate configuration of simulation-networks and selection of methods for process description. Many existing and well-tested modelling principles may also be implemented. This facilitates the use of PINE as a common environment for many well-known models. In PINE, models may furthermore be refined or extended in order to accommodate them to particular simulation needs. For systems that include non-natural components, specialised models may be established. Simulation-methods for the non-natural components may be developed and integrated in the system with minimal efforts.

The growing complexity and interdisciplinarity which are characteristic of water-related problems demand multipurpose simulation tools. PINE meets this requirement since it supports the use of compound simulation variables, and since it allows non-hydrological routines to be integrated with routines for hydrological responses in simulation setups. Furthermore, the simple development of program components for process descriptions makes PINE suitable for exploratory purposes. The fact that users can select process variables to export as simulation results makes model configurations transparent and open to error checking.

So far the PINE program has been tested on simulations of hydrological responses in urban areas and for semi- and fully distributed hydrological simulations of natural catchments. To date, eight students at the Norwegian University of Science and Technology (NTNU) have based their diploma work on PINE. The program system is further in daily use, in several parallel projects, at the research institution SINTEF, which is connected to NTNU. The program will continue to be used and tested at both places.

6. REFERENCES

Abbot, M. and Refsgaard, J.C., Distributed hydrological modelling, *Water Scienceand Technology Library*, Kluwer, 321 p., 1996.

Bergström, S., Development and Application of a Conceptual Runoff Model for Scandinavian Catchments, Ph.D. thesis, Lund Inst. of Technology, 133 p., 1976.

Blöschl, G. and Sivapalan, M., Scale issues in hydrological modelling: *A review, Scale Issues in hydrological modelling*, Wiley & Sons, 9-48,1995.

Killingtveit, Å., Alfredsen, K., and Rinde, T., Anthropogenic Influence on Flood Regimes in Norway – Model Development Strategy in the HYDRA Project, presented on RIBAMOD-workshop in Wallingford UK, 1998.

Kolberg, S., Rinde, T. and Tøfte, L. S., Automatisk kalibrering av hydrologiske modeller (in Norwegian), *SINTEF-report STF22 A99402*, 42 p., 1998.

LAS, GRASSLAND User's Guide for Windows95 and NT – Version 1.1, *Logiciels et Applications Scientifiques inc*, 1996.