ABSTRACT: The increased focus on dam safety makes it necessary to train the responsible personnel in flood handling and management of extreme situations. The River Flood and Accident (RIFA) simulator is a computer based training simulator for flood situations in regulated river systems. The program system is aimed at training the power plant personnel in handling extreme flood situations. Through application of the RIFA simulator the personnel should better understand the river system, how it react to large floods and how this can be controlled by operating the reservoirs and the hydropower plants. Through the RIFA simulator the user will also be subjected to other situations that may appear during the flood situation that requires attention. The RIFA program is currently under development, and this paper outlines the requirements, design considerations and applications of the program system.

1. Background and requirements

Over the latest years there have been an increased focus on safety and procedures to handle floods and accidents in Norwegian hydropower systems. The owners and other interests in the river system will run training sessions to prepare the responsible personnel for the situations that may occur in a critical situation. This kind of training exercise requires much preparation and the cost can be very high. Work was therefore initiated by the Norwegian regulatory association to try to build a computer based training simulator that could be used to complement the large scale training sessions (BKK and Hydropower 1998).

The focus is to create a system that would help the user to understand how the river system works during a flood situation, and to expose the user to various events that may occur during the critical period. The user can operate the system by opening and closing floodgates and changing releases through the power plants and bypass gates. The RIFA simulator should not only be concerned with the flood control itself, but also cover situations that is a direct or indirect consequence of the flood situation. During the course of the simulation many types of situations that requires attention may occur, e.g. flood related damages, warnings on problems with gates and questions from the public or media\(^1\). The user actions are logged and evaluated through a scoring system. In many ways the RIFA simulator will be similar to one of the strategy computer "games" now widely available.

To simulate the runoff and routing in the river system the RIFA simulator is equipped with various simulation methods. Since the main focus of the simulator is the system understanding and the event handling, the available simulation tools are

---

\(^1\) Hereafter referred to as the message system
mostly different types of hydrologic routing and simplified methods of runoff generation.

The program is developed on the Windows NT platform building on the standards for Graphical User interfaces and help systems found there. An important goal of the program design is to make the RIFA simulator adaptive to future developments and user needs. Care is therefore taken to make a system where it is easy to customise it to the users river systems, and where it is necessary to add new simulation tools.

A further description of the requirements can be found in Alfredsen et al. (2000).

2. System structure

The RIFA simulator is built using an object-oriented analysis and design strategy. This provides the necessary mechanisms to achieve the needed modularity, reuse and maintainability that are wanted in a modern software system. The object-oriented structure of the system also provides the flexibility needed in a future customisation of the system to specific user needs. The RIFA simulator has four main parts: The Graphical User Interface (GUI), the game engine, the simulation system and the river system model (Figure 1). The four main parts fulfil the following tasks in the system:

The GUI system gives the user a visual view of the system structure, the inflow and outflow from the different components and the states of all gates and turbines. Through the GUI the user can make inflow forecasts and try out different operational strategies depending on the forecasted inflow. The GUI system will notify the user of messages and let the user respond to these. There will also be a warning function that gives a signal if the water reaches critical level at user defined locations in the river system.

The game engine controls the game part of the system. It handles all communication with the GUI, manages the underlying river system model and controls all communication with the simulation system. The game engine is the core of the message system launching the events the user has to handle and processes the user responses. The game engine is responsible for the log functions that store all user interactions, the development in the river system and the score from the gaming part of the RIFA simulator.

The river system model defines the structure and water flow of the river system. All system states are stored in the river system model, including operational parameters like planned releases.
The simulation system contains the routines for rainfall-runoff and routing calculations.

The GUI system is completely separated from the game engine through a specially designed interface. This prevents GUI specific code from being used in the game engine, which aids in future extension and maintenance of the system.

The river system model and the computational methods are built on top of a general object-oriented framework for the design of applications in hydroinformatics (Alfredsen 1999; Alfredsen and Sæther 2000). Through this framework all necessary classes for defining elements of the river system (structural components), connectivity and data handling are defined. The framework is designed so that simulation methods are completely separate from the structural components, thereby making it simple to add or change the simulation methods at run time.

3. Simulation

Runoff generation from rainfall is done through a set of simplified methods such as a linear tank model (Andersen, Hjukse et al. 1983) or already generated runoff curves. There will also be a possibility of using a version of the HBV-model (Killingtveit and Sælthun 1995) that is included in the base framework. This solution will particularly be interesting for users that already have a calibrated HBV-model for parts or the whole of their catchment.

Routing in reservoirs is done through a mass balance method. For each reservoir with gated outlet the user can control releases by defining the opening of the gate. This can be done through the GUI and it is possible to set a release plan for the entire or a part of the simulation period.

Since the focus of this program is not on detailed simulation of river routing, complex fully dynamic hydraulic routing methods will not be integrated in the RIFA simulator. Through the underlying framework, there is a possibility of integrating such tools on a later stage, and this has been done in other applications of the framework (Alfredsen 2000). The user of the RIFA simulator can select the river routing from a series of hydrologic and simplified hydraulic methods.

At any time the user can forecast inflow for a specified time period. Based on the forecasts and the release strategy the reservoir levels, reservoir releases and river discharge is computed.

The RIFA simulator will run with a minimum time step of one hour. There will be several different methods of time handling. The possibilities will be to use real time, compressed time where each time step is compressed by a user defined factor and user controlled time steps where the user advances the program manually. The last option is specially designed for testing and program learning purposes.

4. The game engine

4.1 Scenario setup

The first version of the River Flood and Accident simulator will be delivered with two sets of data covering two different Norwegian river systems. These can be used directly for training, or they can be used as a foundation for development of data sets for a specific river system.
A set of scenarios controls the simulation, which are a combination of a weather situation and a set of messages, error situations and other disturbances. Scenarios can be developed by the person responsible for the exercise, or they can be created by combining the existing weather situations with the different game plans supplied. There are no limitations on the flood situations covered, everything from Q_{t0} to Q_{dim} should be possible to run in the simulator.

4.2 Message handling

The purpose of the message handling system is to test the user in situations that may happen in a real crisis and to increase the pressure on the user to create a realistic work environment. The messages the user has to respond to will in most cases present problems different from just finding the optimal release strategy to minimise the flooding effect. There are basically two types of messages that can occur in the RIFA simulator. These can be categorised as independent or dependent situations.

Independent situations

The name independent situation implies that the outcome of the situation does not have any impact on future error situations in the regulatory devices in the system. The independent situations can be divided into two different types. The first one is controlled by the states in the river system, and they will be related to a specific location in the river system. An example could be a setup that sends a flood warning if the discharge at the location “Kveldssol” exceeds a certain limit. This will cause a message to be presented to the user:

- “The area Kveldssol is threatened by flooding”

The user will then be presented with several actions that can be taken as a response to the flooding threat:

- “Continue to release water to minimise risk for dambreak”
- “Close gates in the dam to save Kveldssol”
- “Send a message to evacuate the inhabitants”
- “Do nothing at all”

Each of these responses will have an associated score, and some of them may trigger new messages either directly or later on in the game. The messages are organised in a hierarchical tree structure. The organisation of the message tree will be similar to the fault trees used in risk analysis.

The second independent message type is meant to disturb the user (increase the work pressure during e.g. important forecasting). This kind of noise can be e.g. phone calls from media or local inhabitants. In most cases there are no sequence of messages connected to this kind of message.

Dependent situations

The dependent situations differ from the independent situations by the fact that wrong responses may directly cause future error situations in the regulatory devices. An example can be the following message:
The user then has to take action. In this case it will be necessary to do something to prevent the debris from clogging the gates. If the user does not respond correctly, the gates may fail at a later time in the game. The dependent messages can have the same hierarchical structure as the independent messages with new messages triggered directly or after a time period.

The messages will be stored on text files, and they can easily be extended or new situations can be added at need. The trigger point for each message hierarchy is stored in a special scenario file.

4.3 Error situations

There are three typical error situations that may occur during a run of the RIFA simulator.

1. Failure in gates and crested spillways. Gates may either fail when closed or they can fail in open position. Gate failure may be a consequence of the user handling an earlier situation wrong, or it can happen without any warning. A crested spillway may fail due to clogging from debris.
2. Turbines may fall out and thereby lead to reduced releases from the reservoir.
3. Error in prognosis due to errors in the forecasted temperature and precipitation. This may lead to the user taking the wrong decisions on the operational strategy.

The game engine will control the error situations and it will also transfer the error states to the river system model so that they will impact the simulation until the error is resolved. All errors will be reported through the GUI and also stored in the program log.

4.4 Resource Pool

Some of the situations the user has to handle will require resources like people or machinery. In the beginning of the period the user will have access to a fixed number of resources of different types. If the user select to use a resource to handle a situation this will be unavailable for a period of time. The user must therefore take care in the resource allocation not to run out of available resources. The available resources can be configured at start-up.

5. The GUI

The GUI is developed using Microsoft Foundation classes and follows the Microsoft standard for user interfaces. Emphasis will be taken to provide an interface that is familiar to users experienced with other Microsoft Windows based programs. There are two main windows in the system. One shows a system overview with dynamic objects. The reservoir and lake objects will show the water level, and the control points will visually display alarm states (Figure 2). The second window display detailed information about each component and let the user edit data before running a prognosis. The graph and spreadsheet controls will be divided into a historical and a prognosis part, as shown on Figure 3.
6. Development plan

The River Flood and Accident simulator should be finished by March 2001. The Norwegian Energy and Regulatory Association will be responsible of distributing the software. Development news and pre-releases will eventually be available from the SINTEF web page (http://www.sintef.no).

Figure 2 The main window with system overview.

Figure 3 Detailed information window for reservoir Bygdin.
REFERENCES


