Safe Hydroinformatics

R K Price, Research Director, Wallingford Software, Howbery Park, Wallingford, Oxon OX10 8BA,UK (now Professor of Hydroinformatics, IHE, Delft, The Netherlands)
K Ahmad, Senior Lecturer, University of Surrey, Guildford, UK

1. Introduction

Hydroinformatics is a multidisciplinary subject that addresses the management of the aquatic environment using information technology. The task of management is complicated by the uncertainty surrounding many of the scientific processes that are inherent to the aquatic environment and by the number of stakeholders concerned to utilise and preserve the environment as a sustainable resource. Any hydroinformatics system will reflect something of the complexity of the processes and demands of the stakeholders. An important function of a hydroinformatics system is its ability to support its users in helping them to achieve their objectives, namely in managing a particular aspect of the aquatic environment. Inevitably such management involves decision making on the basis of a wide range of information of different types and from various sources. An inexperienced user can easily misunderstand historic data and textual information, mishandle the building, development and application of models, and misinterpret their results. Consequently the opportunities for incorrect conclusions and poor decision making are many. There is a need for better tools that can generate a *safe environment* in which users can be expected to achieve more reliable decisions. This can be effected by supporting the user in his or her decision making. In particular, a decision support system can provide the *framework* for accessing relevant information, marshalling and analysing the data, and making informed decisions.

Besides the need to assimilate data from different sources, the uncertain and incomplete nature of the raw data is such that there are always going to be important questions about the reliability and safety of the decision making based on the data. The problem is exacerbated by the fact that the analysis tools, algorithms and associated modelling systems also contain particular uncertainties, approximations and assumptions that can in extreme cases invalidate the results of their application. Modelling systems, in particular, are dependent for their reliability on correct and proper use in terms of model building, development and application. With the heavy dependence of a hydroinformatics system on data and models the reliability and safe operation of the system is a crucial issue for hydroinformatics. The support for the user is therefore vital. Considerable assistance can be given by highly user-friendly, intuitive interfaces operating with natural language and access to extensive context sensitive help. Such features are important. But whereas these are now widely accepted as being necessary to make the software easy to use there are still many difficulties faced by inexperienced users such that safe use of a hydroinformatics system may be prejudiced.

Inexperienced users of simulation modelling software may assume naively that the software system they are using is error free. Increasingly, however, more informed end-users have begun to ask questions about the quality of the software, its reliability and testing. Such questions are directly relevant to design failure or safety of the physical asset which has been designed or modified using the software system. This aspect of safety is being explored by software engineers and end-users working together in two major disciplines: *safety-related systems* and *safety-critical systems***.** A safety-related system is one whose malfunction, either directly or indirectly, has the potential to lead to safety being compromised, whereas a safety-critical system is one in which any failure or design error has the potential to lead to loss of life.

The literature on safety-related systems and safety-critical systems focuses, by and large, on design and operation in well-publicised, potentially safety-compromising areas such as nuclear engineering, air and space craft design and operation, and health and safety at work. The umbrella term used for discussing safety issues related to IT systems is *safe information technology*. Interestingly, there is little existing literature on how to design and operate water-carrying networks *safely*. These networks on the whole may be regarded as *safety related systems*. However, given the growing public concern about threats to public health and safety from badly designed or poorly operated water-carrying networks, there are instances where some, if not all, of these networks may be classified as safety critical systems.

What we seek is a principled framework within which we can discuss how to synthesise systematically the union of human and artificial agents directed to the design of water-carrying networks and to urban drainage networks in particular. The synthesis of safe information technology methods and practices within such a framework leads to the establishment of *safe hydroinformatics*. By design we mean a knowledge based task that involves almost simultaneous access to a variety of knowledge sources, including analytical, formal, experimental and experience-based sources, and also involves a deft mixture of sophisticated computation, qualitative reasoning and extensive use of rules of thumb (or *heuristics*). This simultaneous access to a variety of knowledge sources is crucial in the design of networks that carry water, gas, oil or electricity.

Apart from major and typically rare catastrophes water networks can be designed and operated safely and efficiently. This is partly due to the fact that networks supplying fresh water and draining away waste water were amongst the first significant engineering artefacts created by society. With access to experiential knowledge and knowledge based on local idiosyncratic conditions pre-Newtonian water engineers were able to drain water effectively from agricultural and urban land, although not as efficiently as their 20th century counterparts. But if there are so few catastrophes, how is it that the designers of water carrying networks, relying largely on heuristics, qualitative reasoning and local knowledge, produce designs that fail so infrequently? One possible answer is that the identification of the physics of fluid flow, the mathematical description of the physics, and the implementation of the description in a software simulation system, is but a small, even if vital, part of the entire decision-making process. In this sense the use of physical theories, mathematical formalisms and computer programs has to be situated in the broader context of such factors as the location and state of the drainage network, intuitive insight into system performance, the characteristics of the conurbation, the history (if any) of the network, its performance in terms of flooding and pollution, and existing management priorities and public health targets. This comprises *knowledge* of the domain. Such knowledge is the major factor in the production of safe designs. In other words, what the designer *does* with his knowledge and available tools is just as, if not more, important as the safety or reliability of the tools themselves. Given, therefore, that knowledge of the domain is crucial there are well known ways of collecting such knowledge systematically from experts and representing it in *knowledge bases* together with relevant documents such that the knowledge can be retrieved by relatively inexperienced engineers to support them in solving specific problems.

In this Chapter we focus on how a computer can mediate in providing decision support to users of complex modelling systems. In particular we explore how to assure safety during various phases of the rehabilitation of a drainage network. We elaborate the various notions that are central to hydroinformatics, namely those of integrating IT methods, tools and techniques in a common framework to address problems related to the aquatic environment. Our emphasis is on the safe design of drainage networks through the use of software engineering techniques, and through the provision of interfaces. Such interfaces are needed to simulation engines used in the design, to expert systems containing knowledge of drainage experts, and to 'digital' libraries of relevant documents. The elaboration is conducted through the description of a recently completed SAFE Design of Networks using Information Systems (SAFE-D*IS*) project.

2 Failures, hazards and risks in sewerage network rehabilitation

A typical engineering application of sewerage network models in the UK is to rehabilitation planning. In 1984 the Water Research Centre published the first version of its Sewerage Rehabilitation Manual (SRM) that incorporates an engineering procedure to facilitate the development of ‘drainage area plans’; see Water Research Centre (1984, 1986, 1995). Such plans involve the presentation and analysis of a number of options, some of which require a range of engineering judgements, some involve public health consideration, and others include cost-benefit analysis. Drainage area plans involve the recognition of priorities for rehabilitation at a regional level. These priorities are incorporated in the asset management plans of the 10 UK regional Water Service companies as required by the Office of Water, which is the UK government regulatory body that sets investment targets for each of the companies.

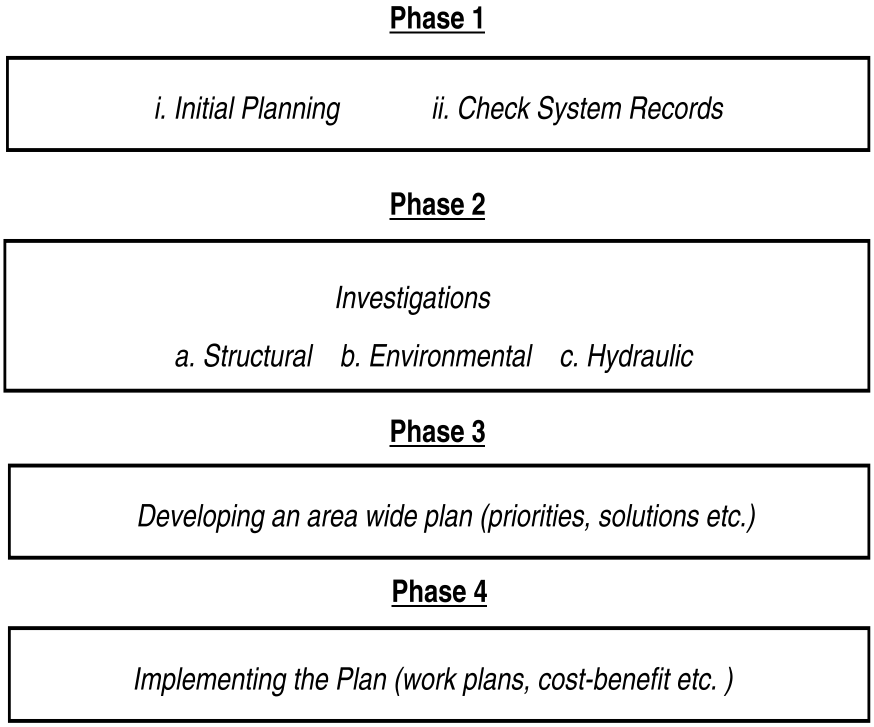


Figure 1. The Sewerage Rehabilitation Manual (SRM) method established by the UK Water Research Centre in consultation with the water industry.

Although not explicitly mentioned, the prevention of hazards and the anticipation of network failures, are amongst the principal considerations of the SRM. Consider, for instance, the key term 'critical sewer' that is used very frequently throughout much of the documentation. 'Criticality' is defined in terms of 'sewers with most significant consequences in the event of structural failure'. A related term is 'core area', which is that part of a sewer network containing the critical sewers and other sewers where hydraulic problems are likely to be most severe and require detailed definition within a flow simulation model. 'Acts of God', in their legal sense, also cause problems, so rehabilitation experts talk about 'catastrophic rainfall event', which is an event of return frequency far in excess of any sewerage design performance criteria, such as a 1 in 20 year storm. Sewer rehabilitation involves monetary expenditure and 'social costs'. Examples of the latter are 'unclaimed business losses due to road closures, and the cost of extended journey times due to traffic diversions'.

Each of the four main phases of the SRM involves a number of considerations about the environmental impact of a rehabilitation scheme. Such considerations are elaborated in terms of 'systems failure', 'hazard prevention', and so on. Tables 1a and 1b comprise the description of various tasks associated with two of the phases of rehabilitation planning. These tasks are annotated with terms like ’failure’, ’hazard’ and ’precaution’ to illustrate the implicit safety issues.

Table 1a. SRM Phase 1: Initial Planning and Records

Phase 1.i	
Task: Determine Performance Requirements	
Hydraulic performance: (failure)	Operational performance: (failure)
Structural integrity: (failure)	Environmental protection: (hazard)
Task: Assess Current Performance	
Use records of flooding (hazards)	
Task: Is full investigation appropriate?	
Full investigation (cost), Structural investigation , Rural investigation	
Task: Check regional priorities	
i) Known causes (failure)	ii) areas of imminent development (precaution)
iii)poor storm sewage overflow (hazard)	iv) system with large number of critical sewers (failure)
v) remaining critical sewers (failure)	
Phase1.ii	
Task: Check System Records	
Depth of sewer; ground quality; marginally important traffic (failure)	
Task: Identify critical sewers	
Collect information (highly impervious - roads) Apply screening procedure (sewer type A, B or C)	
Task: Plan records upgrading and improving access	
Produce Master Plan	

Table 1b. SRM Phase IV: Implementing the Plan

Task: Timing of Construction	**Task: Maintain Hydraulic Model**
OFWAT & Company Rehabilitation Targets (failure) Unit Cost (criticality judgement)	Audit trail must be kept for the model
Task: Timing for Hydraulic work	**Task: Review Drainage Area Plans**
Planned New Developments(precaution) Legislation (hazard)	Major changes New systems coming on-line OFWAT requirements
Task: Design and Construction	**Task: Deal with system failures**
Flooding (failure & hazard) Operational Deficiencies (failure) Structural Condition (failure) New Developments (precaution) Legislation Changes (new hazards) Pollution (hazard) External Influences (failure) Risk	If a collapse occurs; • Make it safe • Carry out repair • Monitor the area If a hydraulic problem occurs; • Develop solution • Record incident • Implement solution • Monitor solution
Task: Monitor condition of critical sewers (failure prediction)	
Sewers must be surveyed and dated	

3. Hazarding safety in modelling the aquatic environment

A key aspect of sewerage rehabilitation is the use of information generated by a computational hydraulic model. Indeed, at the heart of any hydroinformatics system for urban drainage are one or more such models. These models are normally deterministic and therefore designed to replicate the physics of the flow in a particular urban drainage network. Drawing on the discussion of urban drainage modelling in Chapter xxx considerable attention is given in developing commercial software codes for a sequence of physical, chemical and biological processes. These include the correct interpretation of the physics, chemistry or biology, the analytical formulation of the identified laws or principles, the numerical approximations to interpret the analytical equations, and the formal, quality controlled development of the software. But at each of these stages assumptions and approximations are made, ranging from the dissipation of energy due to the boundary friction in the Saint-Venant equations for gradually varying one dimensional flow, to the 4-point or 6-point implicit finite difference formulations of the equations requiring the introduction of a forward weighting to ensure stability of the finite difference solution and inducing an artificial numerical dispersion, to the bugs inherent in the software despite rigorous development techniques. Thus at every level the modelling software is prone to limitations and uncertainties.

The second category of uncertainty involves limitations in the input data to a model. In order to represent the complex geometry of a sewerage network considerable care is needed over what and how data is collected. The rainfall, rainfall-runoff and wastewater inflows have considerable uncertainties in their measurement or prediction. There are problems of interpretation of point rainfall over an area, estimation of domestic wastewater inflows from population density and other data, calibration of the rainfall-runoff model, and so on.

The third category of uncertainty is induced by the user's interpretation of results from the model dependent on his or her understanding of the modelling software. Whereas an expert user will know many of the pitfalls and short-cuts in building, developing and applying a model, there are still many opportunities for the introduction of errors in judgement. The situation is therefore even more precarious for the inexpert or novice user. Water services organisations have recognised this difficult and have devised procedures such as the SRM to provide users with guidelines that help them produce better models.

Given such considerable potential uncertainties in software, data and modelling/engineering procedures it is of concern to managers and users alike that models for particular sewerage networks may not be safe to use and their results are inaccurate and unreliable for decision making. Questions have to be asked, such as: How can such a safe model be developed? What should a manager do to ensure that a model is safe? What can be done to improve the reliable application of information generated by a model? Given the discussion above it should be obvious that there is no simple answer to such questions. Yet there remains considerable concern among users and managers that without some assurance that models and their results are safe to use then there are considerable economic risks that have to be covered through appropriate precautions. It is the minimisation of these risks that is of concern in the remainder of this chapter.

4. Approaches to improving safety

Consider first the minimisation of risk in the application of a particular simulation model. A number of studies have been made of the behaviour of complex computer programs used in high-technology industries such as aerospace, nuclear power, and air-traffic control. These studies have led many software scientists to believe that some software systems are inherently unstable. Despite rigorous development control the complexity of these systems implies that even at the design level the stability of the software implementation cannot be guaranteed. Myers (1986) estimated that there are about three software errors per thousand lines of code in large software systems connected with the US Strategic Defence Initiative (also known as the Star Wars Initiative). If this is the case for defence systems that have millions of dollars invested in them then the question arises as to the frequency of occurrence of errors in comparatively low cost, less frequently inspected and tested civilian systems.

Questions of reliability of software are what have prompted the safety-critical software programmes. Embedded in these programmes are the advocacy of standards. Bowen and Stavridou (1993) referenced 13 such standards for software systems with varied applications, including three standards for military standards in the US and two standards in Europe for software systems developed for the nuclear power and space industries. More recently Glöe and Rabe (1995) have described German National Standards for safety-related systems (see DIN V VDE 801- Principles for computers in safety-related systems published in 1990 and revised in 1994), the International Electrical Commission's standards for safety-related systems (see IEC 65A - Software for computers in the application of industrial safety-related systems published in 1991) and the International Standards Organisation Quality Standards (ISO 9000) for assessing safety-related issues in nuclear power plants, in medical systems (breathing-support machines) and in the control of railway locomotives. These authors note the existence of a number of software tools for testing software systems that help in the confirmation of a software system used as an integral component of a safety-related system.

Safety of software has therefore had to depend on quality controlled development procedures and a large, but not exhaustive, number of tests on the software. This leaves a finite if ‘acceptably small’ risk of failure. Regrettably, even for comparatively simple hydroinformatics modelling software there are no mathematical algorithms that can be used, even in a rudimentary way, to validate the codes.

It is necessary, therefore, to take an alternative view of how to develop safe codes. What can be deduced from the discussion is that software is more likely to be ‘safe’ for use if it is developed by an organisation for whom safety is important, that is, where a quality standard is in use. This should also be the case where the software is to be implemented in the design and assessment of large capital schemes, where the reputation of the organisation has direct financial consequences, and where the scientific and mathematical basis of the software is mature. The European Hydraulic Laboratories recognised the need for some means of assessing the validation of software codes for hydraulic modelling; see IAHR (1994). The approach adopted was to examine each of the main steps in the development process:

- deduction of the relevant science (physics, chemistry, biology)
- development of the mathematical algorithms interpreting the science
- production of numerical algorithms to interpret the mathematics
- formulation of the modelling structure using the numerical algorithms
- production of software code

Safety is hazarded at each of these steps. Therefore some attempt to validate the assumptions and conclusions during the process should be made. Dee (1993) recommends the development of a paper-based dossier in which the arguments in favour of the assumptions and conclusions are documented, including numerical evidence as appropriate. In no way can such a dossier be complete, but it can provide the basis for an assessment by others of the possible risks that may be incurred in using the software.

Given software that has been proven by extensive use and access to reasonably reliable data it remains for the user to carry out the procedure of building, developing and applying a model using the data with the software. It is at this stage that the majority of problems in safety occur. No two people will produce exactly the same model. An expert modeller is more likely to produce a safe model than a novice. But even the expert modeller can make mistakes and be unaware of the risks that his or her decisions generate. Can therefore the reliability of the expert be improved still further? Also, how can the knowledge of the expert be transferred effectively to the novice? These and other questions complement the questions above.

5. Decision support systems

One way of improving safety in making decisions is through the access by engineers to information and knowledge that can support them in their work. The notion is that appropriate expertise is made available through what are loosely termed ‘decision support systems’. These systems may simply provide context sensitive information. They may guide the user through a reasoned process. Alternatively, the user may have the full support of an expert system.

There are a number of examples in recent years of decision support systems in urban drainage. One of the first such systems was WIFE: the WASSP Intelligent Front End (Ahmad et al 1985) where WASSP was a 4th generation modelling system for urban drainage; see Price (1981). Another key example in the UK was SERPES which was developed during the UK Government Alvey programme in the mid-1980s; see WIESC (1988) and Chapter 2. This prototype expert system was focused on an advisor for sewerage rehabilitation planning. It was based on an already existing document: the Sewerage Rehabilitation Manual, produced by WRc and was complemented by extensive interviews with practising engineers. The objective of SERPES was to take the user through the complete process for what was termed drainage area planning. Advice was given on each step of the different sub-processes. Direct links were provided to modelling software which was a necessary tool for the hydraulic analysis phase of the procedure. The expert system was not finally taken up by the industry for a variety of reasons including the expense of developing the full implementation of the product, the uncertainty of the future of the industry with privatisation on the horizon, and the awareness that the tools for building expert systems were not sufficiently stable or flexible. Another important expert system for the selection of the best model to use in a given situation was developed under the EC COMETT programme; see Griffin et al (1993). This involved the development of a tool kit based on an expert system to transfer knowledge and provide training with the use of several computational hydraulic models for urban drainage originating in Europe, namely MOUSE, HYSTEM-EXTRAN, WALLRUS, SPIDA and BEMUS.

A number of other decision support or expert systems have been developed to address different aspects of urban drainage modelling. For example, Liong et al (1991) for a knowledge-based calibration procedure for the SWMM RUNOFF block. The concept involves a sensitivity analysis of the calibration parameters and a strategy for parameter selection that attempts to match the simulated and observed hydrographs. Other similar calibration systems have been developed by Delleur (1991), Delleur and Baufaut (1990) and Baufaut and Delleur (1989, 1990) for calibration of rainfall-runoff and runoff quality modelling. Bowland et al (1993) report on BMP-PLANNER as a decision support system and educational tool for stormwater quality management. This uses models such as XP-AQUALM and SWMM depending on the degree of complexity of modelling. Similarly, Alfakih et al (1990) describe an artificial intelligence system to assist with decision making in integrating alternative solutions for urban drainage problems. A fruitful area for applying decision support systems is real time control. Jacobsen et al (1993) describe the development and application of a general simulator, SAMBA-Control, for rule based control of combined sewer systems, while Lindberg et al (1993) extend the concept further with MOUSE ONLINE. Khelil et al (1993) has also adapted an expert system for the real time control of a sewerage network.

It is important to recognise that although decision support systems offer considerable benefits within a hydroinformatics system they are not infallible and do have some serious limitations. The chief limitation is in the structures that are available for computer-based reasoning.

Human beings reason in a variety of ways that include intuition and the exercise of feelings. Symbolic computer-based reasoning is, in comparison, very limited; see Abbott (1991). Knowledge has to be represented in a structured manner. Reasoning can at best be based on rationality; see Dreyfus and Dreyfus (1989). Therefore, any decision support given by the computer using the symbolic paradigm is limited; see Amdisen (1995) for an exploration of the use of rational reasoning in a hydroinformatics system. Although there is considerable scope here for the introduction of sub-symbolic paradigms, much can be done by recognising that a hydroinformatics system properly *includes its users*. This opens up the possibility of a symbiotic relationship between the user and the computer-based hydroinformatics system that takes advantage of the separate and distinctive ways of reasoning of both human and machine. A very good example of this relationship is shown in a recently completed project that explored the SAFE-Design of networks using Information Systems (SAFE-D*IS*).

6. Safe design of networks using information systems (SAFE-D*IS*) project

This project was a three-year (1993-1996) collaborative venture between a university (Surrey) and a vendor of specialist software systems (Wallingford Software). The project addresses safety related questions regarding the safe design, cost-effective repair and the subsequent hazard-free operation of large *in-situ* drainage networks. These networks serve large conurbations, comprise hundreds if not thousands of conduits (pipes) interconnected through a number of nodes (including inflows, outfalls, pumps, storage tanks), and require significant capital investment to effect changes in their design and subsequent repair or *rehabilitation*.

The SAFE-D*IS* project was joined by the SAFE-D*IS* Round Table consisting of members from the private sector (Thames Water PLC., Severn Trent PLC., and North-West Water PLC), the public sector (Walsall Borough Council, which acts as an agent for the Severn Trent Water for drainage matters in the borough of Walsall, and Sheffield City Council) and a UK civil engineering consultancy, Montgomery Watson PLC. The University of Surrey and Wallingford Software helped with the organisation, execution and follow up of the Round Table meetings.

Knowledge related to the safe and cost-effective rehabilitation was acquired by the SAFE-D*IS* project team from human experts and from specialist texts. The text corpus comprises safety guidelines and procedures, transcripts of expert interviews, learned papers and technical notes, legal texts including the complete Water Resources Act 1991 (HMSO) and a 450 page book that interprets previous legislation; see Wisdom (1970). The text corpus also consists of a terminology data base. All the texts relate in one way or another to the rehabilitation of drainage networks. This knowledge was structured in an information system for facilitating safe and hazard free rehabilitation of a part of the network. The structured knowledge can be used to help experts examine their own knowledge, and assist novices to a greater or lesser degree throughout various phases of the complex rehabilitation process.

The SAFE-D*IS* project has identified five distinct groups of software systems that may help in the five key functions that are essential for the safe rehabilitation of complex drainage networks; see Table 2. The integration of these systems was one of the achievements of the project:

Table 2: Key functions for safe rehabilitation of complex drainage networks

Function	**Software Components**
Access electronic documents	Full-text and hypertext management
Access rules and heuristics	Knowledge-base management
Modelling complex network	Network simulation software
Sensitivity risk analyses	Risk analysis software
Model history and audit	Report generating systems

One of the important decisions of the SAFE-D*IS* project was to use as much off-the-shelf software as was possible without compromising the high standards that are demanded for a safety critical system. Thus the information system has access to proprietary simulation, risk analysis, and text analysis software, knowledge engineering tools and data base management systems. The information system developed by the project team animates the behaviour of an experienced engineer setting a number of tasks for a less-experienced engineer to execute. This animation is based on an industry-wide rehabilitation procedure that involves over 20 specialist tasks distributed over 4 major phases; see Figure 1 above.

The first phase of development in SAFE-D*IS* resulted in a conventional software system. Much like the conventional software system, including database systems or simulation engines, the first prototype relied on the user having sufficient motivation and/or knowledge to use any of the textual archives, the simulation engine, the propositions database or the automated procedures. Thus the system reacted to a knowledgeable user quite well, but for novices and, indeed, some experienced engineers, the operation of the system was somewhat baffling.

6.1 A knowledge-rich, integrated, proactive safety information system

SAFE-D*IS* is a *proactive* system, that is, a system that can execute the major and ancillary tasks outlined in each of the four major phases of sewerage rehabilitation planning This proactive system acts in many ways like other proactive systems, for instance, an expert system, wherein the system infers new facts from old, depending on the context, looks up and presents data from diverse sources, invokes other software systems, and so forth.

This proactive system acts within the framework of the SRM method (WRc (1986)). During the execution of individual phases, and tasks within a phase, the proactive system provides *expert advice*, based on rules of thumb and other heuristics obtained from experts. Proactively, the system can access excerpts and (optionally) full-text from a 'corpus' of texts, some of which are linked through hypertext links. This provides a digital library built in close collaboration with the Round Table. Advice is supplemented by access to data bases containing details of the various components of a given network and its geographical location, and supplemented by access to a industry-standard simulation model, namely *HydroWorks*, developed and marketed by Wallingford Software (1994).

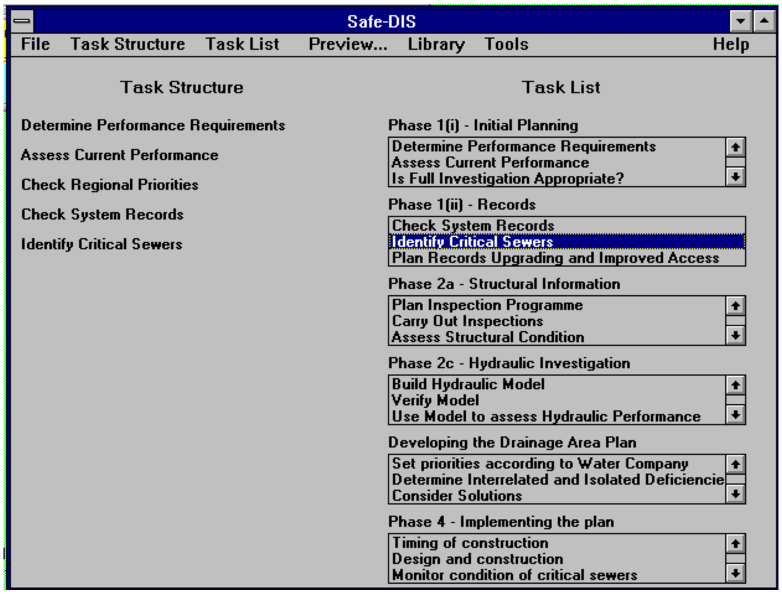
The system also keeps a **'**diary' of advice it gives to a user and invites the user to enter his or her comments on the advice given or to justify a particular decision that is deemed by the system to be out of the norm. Risk analysis, an important tool in the safety community, can be undertaken through the information system using a low-cost, easy-to-use, and off-the-shelf system, namely, Crystal Ball^{TM} marketed by Decision Engineering Ltd. The information system also provides access to the World-Wide Web and through the Web provides access to up-to-date information related to engineering, legal and safety aspects of the aquatic environment as and when it becomes available on the Web; see Table 3 for more details. More advanced users of the information system have access to a text analysis system, namely *System Quirk*; see Ahmad and Holmes-Higgin (1995) and Chapter 2.

Table 3. The functionality of the various components of SAFE-D*IS*. The user interface of the Workbench is written in Visual Basic and runs on a PC. The knowledge-bases are encoded in a variant of PROLOG.

Software Component	**Narrative**
Task Selection & Display	Enables an expert/manager to select tasks for a given project to be executed by a novice engineer.
Knowledge Management	Manages the knowledge base of the SAFE-D*IS* system and contains rules related to various rehabilitation tasks
Yellow Pages Management	Tracks the task a given user is executing and selects relevant excerpts (paragraphs and pages) from a full text-data base.
Safety Labels Management	Displays 'safety labels' during or after the execution of a rehabilitation task
Diary Management	Tracks when and how successfully each task was executed and notes it in a diary.
Report Generation	Generates an 'audit' report based on the contents of the 'diary'
Plug-In External Software	Helps to access data in proprietary data bases and acts as a front end for simulation software.

6.2 Operational Details of the SAFE-DIS Workbench

The SAFE-*DIS* workbench offers two modes of operation: *professional* and *roster*. The professional edition refers to a mode of operation designed for experts where they can either browse through the system, add more knowledge, modify or delete existing knowledge, and select some or all the phases, and tasks within the phases, for execution by less-experienced engineers. SAFE-D*IS* can thus be configured by senior design engineers in two important respects.



Task Structure

Determine Performance Requirements
Assess Current Performance
Check Regional Priorities
Check System Records
Identify Critical Sewers

Task List

Phase 1(i) - Initial Planning
Determine Performance Requirements
Assess Current Performance
Is Full Investigation Appropriate?

Phase 1(ii) - Records
Check System Records
Identify Critical Sewers
Plan Records Upgrading and Improved Access

Phase 2a - Structural Information
Plan Inspection Programme
Carry Out Inspections
Assess Structural Condition

Phase 2c - Hydraulic Investigation
Build Hydraulic Model
Verify Model
Use Model to assess Hydraulic Performance

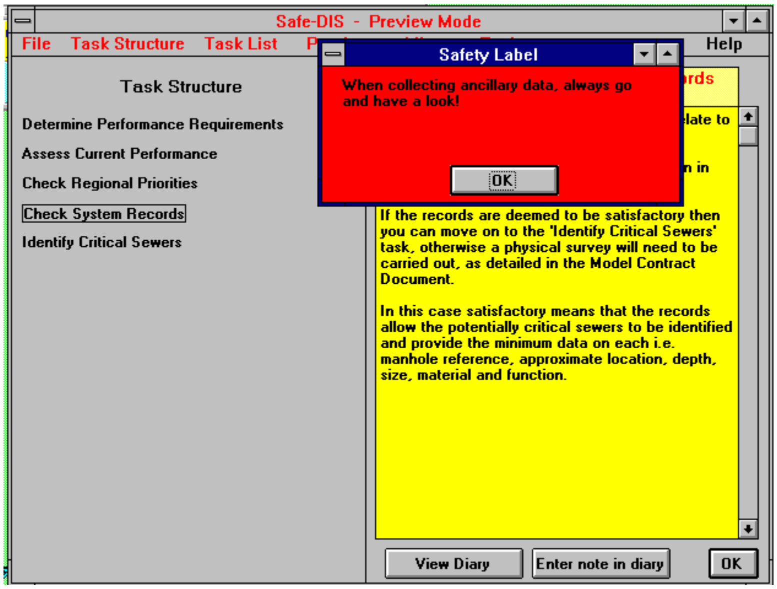
Developing the Drainage Area Plan
Set priorities according to Water Company
Determine Interrelated and Isolated Deficiencie
Consider Solutions

Phase 4 - Implementing the plan
Timing of construction
Design and construction
Monitor condition of critical sewers

Fig 2: Working screen of SAFE-D*IS*

The first level of (re-)configuration is at the knowledge-levels whereby a designated user can add or delete sub-tasks to any of the four phases of the SRM method. The second level of configuration is one where the senior engineer selects specific sub-tasks from one or all the four phases which he or she thinks should be investigated by one or more engineers reported to him or her.

The roster edition refers the operation of the system by novice engineers where advice is provided and the they can browse through the text corpus and access databases and simulation models. During the execution of each of the rehabilitation task, the user of the system is guided through a question and answer session that includes display of *safety labels* containing brief items of advice.



Task Structure

Determine Performance Requirements
Assess Current Performance
Check Regional Priorities
Check System Records
Identify Critical Sewers

When collecting ancillary data, always go and have a look!

rds

late to

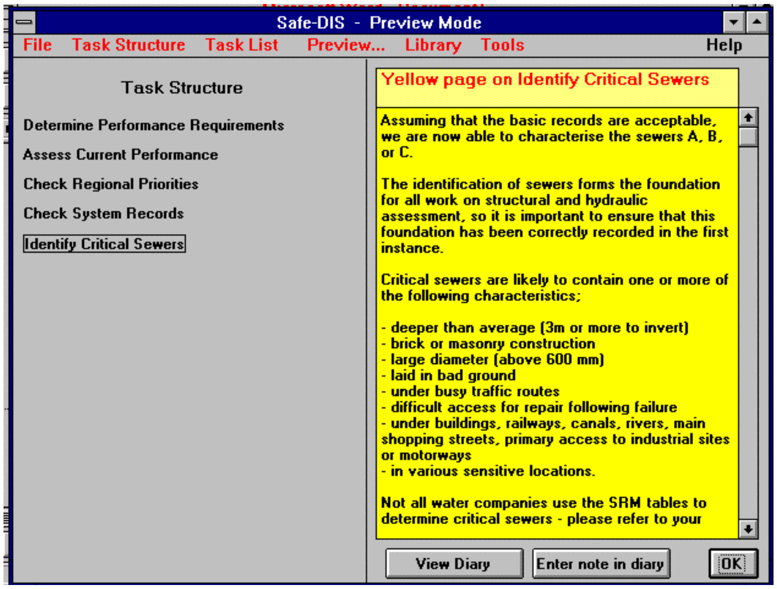
n in

If the records are deemed to be satisfactory then you can move on to the 'Identify Critical Sewers' task, otherwise a physical survey will need to be carried out, as detailed in the Model Contract Document.

In this case satisfactory means that the records allow the potentially critical sewers to be identified and provide the minimum data on each i.e. manhole reference, approximate location, depth, size, material and function.

Fig 3: Safety Labels

During the interaction the workbench provides pro-active advice: excerpts of texts shown in so-called *Yellow Pages*. Safety labels are sometimes displayed concurrently with the Yellow Pages. The labels come in three 'colours': *red* for mandatory warnings; *amber* for potential hazards; and *green* for safety notes.



Yellow page on Identify Critical Sewers

Assuming that the basic records are acceptable, we are now able to characterise the sewers A, B, or C.

The identification of sewers forms the foundation for all work on structural and hydraulic assessment, so it is important to ensure that this foundation has been correctly recorded in the first instance.

Critical sewers are likely to contain one or more of the following characteristics;

- deeper than average (3m or more to invert)
- brick or masonry construction
- large diameter (above 600 mm)
- laid in bad ground
- under busy traffic routes
- difficult access for repair following failure
- under buildings, railways, canals, rivers, main shopping streets, primary access to industrial sites or motorways
- in various sensitive locations.

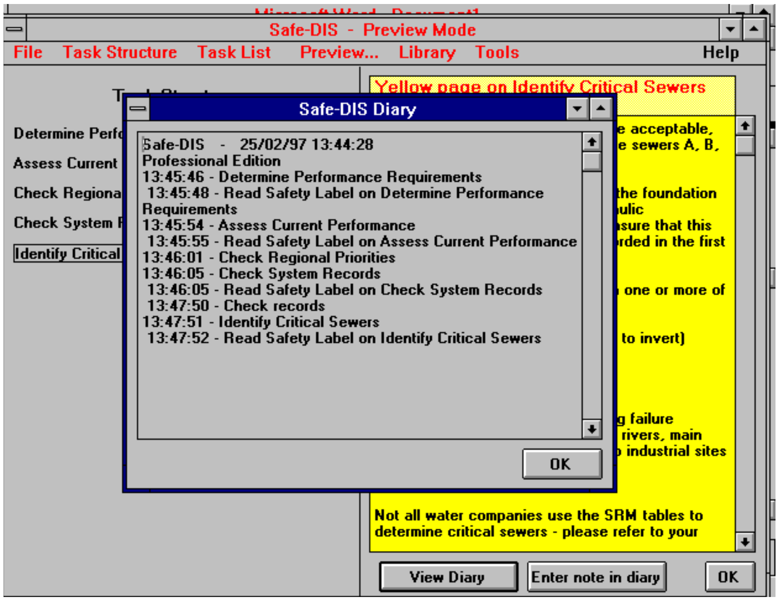
Not all water companies use the SRM tables to determine critical sewers - please refer to your

Fig 4: Yellow pages

The access to full documentation, including Technical Notes (about 10 in number) authored by leading rehabilitation experts in the UK together with expert interviews and legislation is also provided by the workbench.

6.3 Report Generation and Auditing

The end of the interaction with SAFE-D*IS* is marked by the generation of a 'sessions report' for the end-user and, where appropriate, for his or her manager.



```
Safe-DIS   -   25/02/97 13:44:28
Professional Edition
13:45:46 - Determine Performance Requirements
 13:45:48 - Read Safety Label on Determine Performance
Requirements
13:45:54 - Assess Current Performance
 13:45:55 - Read Safety Label on Assess Current Performance
13:46:01 - Check Regional Priorities
13:46:05 - Check System Records
 13:46:05 - Read Safety Label on Check System Records
 13:47:50 - Check records
13:47:51 - Identify Critical Sewers
 13:47:52 - Read Safety Label on Identify Critical Sewers
```

Yellow page on Identify Critical Sewers

e acceptable,
e sewers A, B,

the foundation
ulic
sure that this
rded in the first

one or more of

to invert)

g failure
rivers, main
industrial sites

Not all water companies use the SRM tables to
determine critical sewers - please refer to your

Fig 5: Report generator

6.4 Simulation Engines

One of the subsidiary objectives of the SAFE-DIS project was to investigate the feasibility of intelligent front-ends and another was to investigate how heuristics and rules of thumb may be introduced in the development of simulation engines; see Ahmad (1995) for more details. In the initial stages of the project it was thought that SAFE-D*IS* will essentially provide an intelligent front-end for HydroWorks, that is, an intelligent system to aid in the selection of data for the simulation engine and in the interpretation of the output produced by the engine.

HydroWorks, like other simulation engines such as SWMM-EXTRAN and MOUSE, appears to be adapting and incorporating a number of data management features such as improved data handling and visualisation, and there is good control of software releases. The vendors of HydroWorks, Wallingford Software, and of MOUSE, Danish Hydraulic Institute, are also taking on board notions like quality management of the modelling process itself, including audit trailing and the generation of reports. This implicit development of an intelligent front-end undertaken by these vendors is a welcome development and has helped the SAFE-D*IS* project team to focus on safety-related aspects of the modelling process itself.

7 Knowledge Documentation: The role of the ‘Round Table’

The project used a number of knowledge acquisition techniques reported in the artificial intelligence literature including face-to-face video taped interviews, structured walk-throughs, questionnaires, and interactive rule elicitation; see for instance Boose (1992) and references therein. Face-to-face interviews between experts and system builders were held on topics related to the safe rehabilitation of networks based on a case study. The questions in the interview were devised by the Round Table. Each interview was video-taped and the transcript of each interview was discussed by the Round Table during brainstorming sessions. The system builders extracted specialist terminology from the interviews, and extracted heuristics and rules. The transcript was marked up such that key parts of the interview could be extracted and linked to other documents through a hypertext browser.

7.1 Knowledge acquisition techniques

A summary of the techniques used are given in table 4:

Table 4: Knowledge acquisition techniques

Informal or overview interviews	These are aimed at familiarising the knowledge engineer with the domain and the particular problem which the proposed expert system is intended to solve. It is therefore likely to be the first interview session held with the domain expert and requires much preparation by the knowledge engineer in collating and learning the relevant technical terminology	This serves two main purposes, firstly, to ensure that the knowledge engineer can understand what the expert is saying, and secondly, to enable him/her to ask intelligent questions referring to an object using the correct term.
The structured interview	Structured interviews normally occur well into the knowledge acquisition phase. They are used when information is required in much greater depth and detail than the other techniques can offer and is more interrogative than conversational. The knowledge engineer will have prepared a list of topic headings rather than questions with which to conduct the interview. He/she proceeds by stating his/her understanding of a topic, his/her exact information needs and will prompt the expert to answer by asking a broad question. During the expert's answer the interviewer will regularly prompt for detail, or tactfully interrupt if the information he/she seeks is not being delivered or if the information is too detailed. In this case the interruption might be a request to briefly recap, or to repeat the description of the situation using layman's terms.	The principle outcome of the structured interviews are the details of the domain entities (tasks, rules and objects) to such a level that a decision could be made about the representation scheme or data structures required to implement them in an expert system.
"Think aloud" protocols	This technique has its origins in cognitive psychology where it was used by psychologists to study the strategies with which people solve problems. Knowledge engineers use this technique in the same way though their subjects are generally of similar intelligence and abilities and the problems they are attempting to solve are far more complex. Basically, it requires that the expert 'thinks aloud' while solving a given problem or case study.	Case studies are advantageous because the end results are already known so the expert should repeat the strategy he/she used for that problem when describing his/her solution.

7.2 Brainstorming

The use of brainstorming techniques is seldom reported in the knowledge acquisition literature, yet the technique turned out to be very useful for devising questionnaires for the interviews, and subsequently for validating and verifying the acquired knowledge. In the SAFE-D*IS* Project, the brainstorming sessions were focused on the safety aspects of the specific phases of the rehabilitation procedures; see Figure 1 above. Individual members of the Round Table were given responsibility for providing knowledge related to given tasks in a specific phase; a detailed transcript of each of the sessions was prepared and circulated to the other members.

Corrections and modifications to the transcripts of the interviews and the brainstorming sessions were agreed by the Round Table as a whole. This consensus enabled the system builders to use verified and validated knowledge rather than the (un-revised) knowledge of a single expert as is the case in many knowledge-based systems' projects.

Structured walk-throughs helped in establishing the manner in which the various tasks within a phase are to be structured and in adding more knowledge for a task which the SAFE-D*IS* system could already execute.

Rule elicitation was used to develop automated/standardised procedures. These procedures, mini knowledge-bases, are particularly useful where the task is amenable to formal description, then automating according to a procedure agreed upon by experts will improve safety. During the structured walk-throughs the engineers provided rules and algorithms for various stages of the modelling process, e.g. choosing coefficients of discharge, accounting for unmodelled storage and checking for limits when doing catchment breakdown.

7.3 Knowledge validation and User Testing

7.3.1 Knowledge Validation

The knowledge acquired from these meetings formed the main structure of the Safe-D*IS* system, and was especially used for the yellow pages, introducing each task. After initial transcriptions of the meetings were sent to the relevant experts for their comments, we received their corrections and validations. Any corrections and alterations were made, and the knowledge was fed into the Safe-D*IS* system. It is hoped that the engineers using the Safe-D*IS* system will be able to perform a second level of validation, by commenting and correcting phases which were not there own.

At Surrey we have developed a methodology for the validation of terminology and, as in this case, knowledge. Any new knowledge which is acquired is given a status of R (Red), warning anyone viewing the knowledge to *Stop* and be aware that it is un-validated knowledge. Once the knowledge has been validated by the first expert, it is given a status of A (Amber), implying that the knowledge has been checked once but that the user should *Proceed with caution*. After the knowledge has been validated by a second expert it can be given a G (Green) status, determining that the user can *Go*, as the knowledge is safe to use.

7.3.2 User Testing and Debugging

The SAFE-D*IS* Project team held four workshops at the offices of the Round Table members, in the final year of the Project. Each of the day-long workshops comprised a presentation of the SAFE-D*IS* project and a demonstration of the system to audiences of company personnel ranging from new recruits to senior management. The presentations and demonstrations were followed by open sessions whereby attendees could come along and get a hands-on trial of the system and speak with the SAFE-D*IS* team. The day then closed with discussions which provided further feedback from potential end-users.

Each workshop was attended by over 20 attendees. By conducting the workshops during the life-time of the project it was possible to incorporate changes to the SAFE-D*IS* workbench. Indeed, these visits convinced the project team that what was required was a proactive system, rather than a reactive system, where a user is guided by domain-specific dialogue.

System testing was concentrated on ensuring that the system was sufficiently portable to run, fully, on a variety of different machines. With the introduction of Windows 95 and the increased popularity of Windows NT, it can no longer be said that Windows 3.1 is a standard for Windows users. Thus it was felt that it was important to ensure that the Safe-D*IS* system could run properly on these platforms.

The initial testing involved a user testing the system to destruction, and noting not only any times that the system crashed, but also any strange behaviour within the system. The user involved in this stage of the testing had previously been unfamiliar with the system, and was felt to be sufficiently detached from the project to be deemed to not be biased in any way. This initial testing phase ran for approximately two days, before the comments were collected and modifications to the system were made. These modifications ranged from correcting spelling mistakes in the text, to fixing problems in which the system would crash if a user performed a set of tasks in an unexpected order.

After these modifications had been made the system was tested for a further day, and again any modifications required were made. This cycle continued until it was felt that the system was working as well as possible.

8. Resume and end notes

The design of a complex artefact, like a drainage network, requires careful consideration of number of interdependent knowledge sources, engineering hydraulics and design, environmental sciences, geomorphological and financial data and legal information together with simulation and modelling heuristics. The design engineer works within a community that includes administrators, scientists, and lawyers as well as other engineers. Whether novice or expert, the design engineer should be aware of his or her limitations and seek to compensate for them by co-operating with other expert members of the community.

The implementation of hydroinformatics systems must pay more attention to ensuring that they are used safely. Emphasis should therefore be given to *how* hydroinformatics tools are used as well as to the reliability of the tools themselves. Many of the applications of computational hydraulic models that are at the heart of hydroinformatics systems may prejudice the safety of engineering decisions. Hydroinformatics systems should therefore provide decision support systems that create an environment in which the user is more likely to make informed and reliable decisions.

This issue has been addressed in the development of SAFE-D*IS*. The concepts in Safe-D*IS* emerged during a project at Wallingford Software and the University of Surrey, funded by the UK Department of Trade and Industry under their Safety-Critical Software Programme and by EPSRC. The results are being transferred into HydroWorks for safe design and analysis of urban drainage systems. However, the tools and techniques are generic and can be applied to other modelling or procedural environments. They form a basis for safety-critical hydroinformatics systems.

8.1 The cost-benefits of the safe-DIS System

Wallingford Software has produced a detailed report of the production costs and downstream benefits of the system; see Price (1996). Here we will restrict ourselves to a brief summary of implementation costs and a short description of the implementation vehicle.

The production costs report discusses the costs involved in developing the Safe-D*IS* further into a fully working commercial system. Table 5a summarises the costs involved for the main components required by such a system.

Table 5a - Summary of costs for the main components

Task	**Cost**	
	Fiscal (£k)	**Labour (person days)**
Task Management	28	85
Document Management	18	55
Expert System Shell	5	--
Risk Analysis	6	20
Model Audit facility	11.4	38
TOTAL	**68.4**	**198**

It has been estimated that in order to implement the infrastructure such that it can be used in a design the following extra costs will be incurred. These relate to (a) task objects (b) document markup / analysis and (c) expert system modules. Table 5b contains unit costs for each of the items (a) - (c).

Table 5b - Unit costs for task objects, documents and ES modules

Task	Cost	
	Fiscal (£k)	Labour (person days)
Task object	3	10
Document markup / analysis	1.5	5
Expert system module	25	85
TOTAL	**29.5**	**100**

It has been estimated that 10 task objects and 10 documents will be required together with 3 expert system modules. Thus a realistic estimate shows that to build and implement SAFE-D*IS* will require a total of around 600 person days and £200k.

8.2 Further systems development

The current system will be better implemented using object orientation; this would appear to be the most logical path for any further systems development to take. Each task in the system could be expressed as an object and contain the following slots;

- Texts
- Tools
- Database entries
- Sub-tasks

A task would have a number of texts associated with it, including safety legislation, safety labels, company manuals (or links to relevant sections of manuals), etc. Such tasks would reflect, for example, those in the SRM as described above. However, the concept of tasks is applicable at different levels. For example, a list of basic tasks is given by the pull down menus of the introductory window for the HydroWorks workbench; see Table 6.

Table 6: Tasks as defined in the current introductory window of the HydroWorks Workbench

File	**Edit**	**View**	**Project**	**Model**	**Tools**	**Window**	**Help**
New	Undelete	Select	New	Run new simulation	Generate rainfall	Tile	Contents
Open	Cut	Zoom in	Open	Results	Wastewater generator	Cascade	Search
Close	Copy	Zoom out	Close		Capital costs model	Arrange icons	How to
Close all	Paste	Centre	Files				Tutorials
Save	Delete	Reset	Information				Engineering guide
Save as	Auto insert mode	Options					File reference
Export to DXF	Edit field	Replay					About
Audit	Edit record	Graph					
Print	Insert before	Key					
Print Setup	Find	Find					
Exit	Replace	Select Gauges					
	Validate	Clear Labels					
	Next error	Edit Long					

		Section					
	Previous error	Reverse Long Section					
	Goto	Refresh view					

These basic tasks reflect the need to manipulate data, fire of runs of the engine and to interpret results. At a higher level there are (engineering) process tasks that include various aspects of implementing the SRM or other similar procedures; see Table 7.

Table 7: Tasks at the process level

Model Construction	**Model Development**	**Model Application**
Asset data acquisition	Runoff calibration	Base performance analysis
Critical sewer identification	Asset data confirmation	Performance optimisation
Storage compensation	Sensitivity analysis	RTC design and analysis
Network simplification	Uncertainty analysis	Trade waste analysis
Rainfall-runoff parameterisation	Prototype testing	CSO analysis
Historical performance assessment	Training and implementation of neural network sub-model	Infiltration analysis
Performance requirements determination	Dry weather flow calibration	Flooding analysis
Above ground data acquisition		Sedimentation analysis
Structural condition assessment		Prescriptive network design and analysis
		Storage tank/pond design and analysis
		Capital costs analysis

It is our opinion that the implementation of these and other, similar tasks as objects in frameworks, such as those provided by CASE management tools, will lead to significant developments in *safe* hydroinformatics systems.

9. References

Abbott, M B, (1991), *Hydroinformatics: Information technology and the aquatic environment*. Avebury Technical, Aldershot, UK, p 145

Ahmad, K, Langdon, A, Moss, W D, and Price, R K, (1985), Implementation issues of Hydrological Expert Systems - A Civil Engineering Case Study. In Topping, B (ed), *Proc. 2nd Eng. Comp. Conf. 2,* pp 407-414

Ahmad, K, (1995), A knowledge based approach to the safe design of networks. In Redmill, F and Anderson, T (eds) *Achievement and Assurance of Safety, Proc. of the Safety-Critical Symp.,* London: Springer-Verlag, pp 290-301

Ahmad, K and Holmes-Higgin, P, (1995), System Quirk: A unified approach to text and terminology, *Proc. 3rd Terminology Network Symp.*, Vienna: Int. Network of Terminology, pp 181-194

Alfakih, E, Rarraud, S, Seguin, D and Cao, D, (1993), Global approach to the integration of alternative solutions to urban storm drainage: an artificial intelligence system to help decision making, *Proc. 56th Int. Conf. on Urban Storm Drainage*, Osaka University, p 1341

Amdisen, L K, (1994), An architecture for hydroinformatic systems based on rational reasoning, *J. Hydr. Res*. Vol 32, (Extra Issue), pp 183-194

Baffaut, C, and Delleur, J W, (1989), Expert system for calibrating SWMM, *Jour. Water Resources Planning and Management*, ASCE, 115,3, 278

Baffaut, C, and Delleur, J W, (1990), Calibration of SWMM Runoff Quality Model with expert system, *Jour. Water Resources Planning and Management*, ASCE, 116,2, 147

Boose, J H, (1992), Knowledge acquisition, in *Encyclopaedia of Artificial Intelligence*, Vol 1, ed Shapiro, S C, Wiley-Interscience, New York, pp 719-742

Bowen, J and Stavridou, V, (1993), Safety-critical systems, formal methods and standards, *Software Engineering Journal*, pp 189-209

Bowland, H, Duvinage, M, Goyen, A, and Thompson, G, (1993), BMP Planner: a decision support and educational tool for stormwater quality management, *Proc. 6th Int. Conf. on Urban Storm Drainage*, Niagara Falls, Seapoint Publishing, p 1326

Dee, D P, (1993), A framework for the validation of generic computational models, *Tech. Report X109*, Delft Hydraulics, PO Box 177, 2600MH Delft, The Netherlands

Delleur, J W, (1991), Expert systems in hydrology, *Proc. 2nd Int. Conf. on Computer Methods in Water Resources*, Marrakech, Morocco, Computational Mechanics, Southampton, UK

Delleur, J W, and Baffaut, C, (1993), An expert system for urban runoff quality modelling, *Proc. 56th Int. Conf. on Urban Storm Drainage*, Osaka University, p 1323

Dreyfus, H L, and Dreyfus, S E, (1989), *Mind over machine: The power of human intuition and expertise in the era of the computer*, Basis Basil Black Well

Glöe, G, and Rabe, G, (1995), Current practice in verification, validation and licensing of safety critical systems - The assessor's point of view, in (Eds) Redmill and Anderson, *Achievement and Assurance of Safety: Proc Safety Critical Systems Symp*., London, pp 188-206

Griffin, S, Bauwens, W, and Ahmad, K, (1993), UDMIA: Urban Drainage Modelling Intelligent Assistant, *Proc. 6th Int. Conf. on Urban Storm Drainage*, Niagara Falls, Seapoint Publishing, p 1314

Her Majesty's Stationery Office, (1991), *Water Act*, HMSO

International Association for Hydraulics Research, (1994), *Guidelines for documenting the validity of computational modelling software*, IAHR, Ed. D Dee, 22p

International Electricity Council, *IEC 654*

International Standards Organisation, *ISO 9000*

Jacobsen, P, Hansen, O B, and Harremos, P, (1993), Development and application of a general simulator for rule based control of combined sewer systems, *Proc. 6th Int. Conf. on Urban Storm Drainage*, Niagara Falls, Seapoint Publishing, p 1357

Khelil, A, and Grottker, M, and Semke, M, (1993), Adaptation of an expert system for the real time control of a sewerage network: Case of Bremen Left Side of the Weser, *Proc. 56th Int. Conf. on Urban Storm Drainage*, Osaka University, p 1329

Liong, S Y, Chan, W T, and Lum, L H, (1991), Knowledge-based system for SWMM runoff component calibration, *Jour. of Water Resources Planning and Management*, ASCE, 117, 5, p 507-524

Lindberg, S, Nielsen, J B, and Green, M J, (1993), A European concept for real time control of sewer systems, *Proc. 6th Int. Conf. on Urban Storm Drainage*, Niagara Falls, Seapoint Publishing, p 1363

Myers, W, (1986), Can software for the strategic defence initiative ever be error free?, *IEEE Computer*, Vol 19, pp 11-15

Price, R K, (1981), The Wallingford Procedure design and analysis package, *Proc. 2nd Int. Conf. on Urban Storm Drainage*, Illinois

Price, R K, (1996), *Safe-DIS: System Production Costs*, Wallingford Software

Wallingford Software, (1994), *HydroWorks*, Wallingford, UK

Water Industry Expert Systems Club (WIESC), (1988), *Final Report*, WRc Swindon, UK

Water Research Centre, (1984 - 1st ed), (1986 - 2nd ed), (1995 - 3rd ed), *Sewerage Rehabilitation Manual*, WRc, UK

Wisdom, A S, (1970), *The Law of rivers and watercourses*, Shaw and Sons, London