Distributed environmental impact assessment using Internet

Huang Yan, Dimitri P. Solomatine, Slavco Velickov and Michael B. Abbott

ABSTRACT

The ever more widespread use of the Internet now makes it possible to bring many more persons than hitherto into environmental impact assessment and resulting decision-making processes. Because most at these persons are non-experts, however, it is necessary to provide them with tools that will support their assessments and decision-making efforts. When these tools are directed primarily to the making of judgements they may be described as *judgement engines*. The need to promote cooperative attitudes among participants in the assessment and judgemental/decision-making process, requires that these tools should promote *transparency*. Judgemental processes are introduced and related to evaluation processes so as to provide a characterisation of transparency.

This paper gives an overview of the relevant Internet technologies and then takes the reader through the conception and realisation of one client–server component of an Internet-distributed judgement engine for environmental impact assessment. Because this is built upon the *MikeImpact* judgement engine of the Danish Hydraulic Institute, it is called a *Web-MikeImpact*. Although possibly of interest to specialists in information and control technologies, this paper is primarily intended as a background for potential users of Web-MikeImpact. It should be used alongside the use of the artefact that it describes, as this is available on *http://www.hi.ihe.nl/hi/test/mikeimpact/mikeindex1.htm*.

Key words | environmental impact, internet-distributed decision making

Huang Yan
Forest Department, Bureau of Hydrology, CWRC, Jiefang Avenue 1555#, 430010, Wuhan, China
E-mail: *meiindhi@hotmail.com*

Dimitri P. Solomatine
Slavco Velickov
Michael B. Abbott
International Institute for Infrastructural, Hydraulic and Environmental Engineering—IHE, Westvest 7, 2601 DA Delft, The Netherlands
E-mails: *mba@ihe.nl; sol@ihe.nl; velic@ihe.nl*

INTRODUCTION

Environmental Impact Assessment (EIA) has traditionally been the prerogative of small groups of experts. Correspondingly, the tools that have been provided to expedite EIA processes have usually been configured exclusively for the use of such experts and groups of experts. With the rapidly ongoing development of the Internet, however, many other persons now wish to make their own assessments, whether as individuals or as interest groups. They are persons who, although for the most part not experts, are vitally interested in and concerned about the environmental impacts of certain interventions in nature. They are particularly concerned about the impacts of such interventions upon the qualities of their lives (Abbott & Jonoski 1998). So that non-experts may appreciate the nature of the interventions in nature with which they are confronted and so that they may make realistic estimates of the effect of these interventions on their own life quality, these non-experts need to be supported by a new class of tools. Since these tools are concerned with the making of judgements on the basis of *facts* provided for the most part by existing tools, such as those developed for such purposes as both data collection and processing and for modelling, they are called *judgement engines*. The tools that are used for assembling 'the facts of the matter' for the use of judgement engines are then called *fact engines*.

Processes that involve the making of judgements can be exceedingly various, complicated and convoluted. One possible representation of such processes proceeds through strings of implications, such as:

actions (decisions (judgements (positions (attitudes (beliefs, facts(data)))))). (1a)

In such a case, any number of feedback loops may exist, and specifically data may intervene elsewhere then in its interaction with beliefs, while other complications commonly arise. In such simplified schemata, we subsume all such feedback processes that arise during the judgemental process under the one rubric of *experience*. In the event that implications are treated as mappings, processes of this kind can be represented using the arrow notation of category theory as (Abbott & Dibike 1998*a*, *b*):

(Beliefs, facts(data))→attitudes→positions→ judgements→decisions→actions. (1b)

It should be emphasised that in category theory, as developed in mathematics, the objects linked by arrows must necessarily be mathematical objects, whereas in the present case these are quite other, and more general, conceptual objects. Since the time of Brentano (e.g. 1862/1960) they have been most commonly characterised as *intentional objects.* In the general theory of objects of Meinong (e.g. 1913) they are not even strictly speaking objects (*Gegenstände*) but 'objectives' (*Objekten*). The nature of structures of the kind of (1) have of course been studied since time immemorial and within many contexts, but for our present purposes it may suffice to consider them exclusively within the context of phenomenology under the one rubric of *intentionality* (Husserl, e.g. 1938/1957). Within this context, the role of a judgement engine can be construed as one of providing a specific structure of the kind of (1) around the nexus of judgements. It thus provides a structure that is common to the many participants in the decision-making process. In the analogy to category theory, the role of a judgement engine is precisely this: to constitute implication strings of the type of (1) that are congruent over a wide class of participants, so that the set of all such strings taken over all such participants itself comes to constitute a category in the specific sense of congruence of structures of category theory.

In this same vein, the word 'judgement' is used here in one particular way, as appertaining to judgements made deliberately and logically so as to realise a specific intention. Naturally, acts of judgements occur along the entire string of (1), but these are usually of a more general nature and are less deliberate, less consciously logical, and indeed are often quite instinctual (see Husserl 1938//1973, sections 4 to 11).

Taking this analogy to mathematical constructions further, the culminating result of the life work of Meinong, showing the congruence of the theory of objects (*Gegenstandstheorie*) and the theory of values (*Werttheorie)*, corresponds to the postulate of a *functor* that maps a category of (intentional) objects into a category of values (and vice versa).

We observe, in concluding this introduction, that reversing the directions of the arrows in (1b) corresponds to another process again, in which we attempt to deduce the judgements, and thus the positions, and thus the attitudes, and finally the beliefs of other persons, from their actions and our knowledge of the facts with which the beliefs interact:

Actions→decisions→judgements→positions→attitudes→ (beliefs, facts(data)). (2)

In this case, actions and facts (data) play the roles of observables.

This provides in any particular case a category that is the dual of the original category represented by (1b). A corresponding category of values may then be either a covariant or a contravariant functor, depending upon whether it maps from the category characterised by (1b) or from the category characterised by (2). The class of judgement engines with which we are concerned here is one that facilitates the making of judgements in the sense of (1) and which may be extended subsequently in order to comprehend processes occurring in the sense of (2). It is thus expected, in time, to provide a functor-like capability, so that mappings into processes of valuation, or 'evaluation processes' in the strict sense, can be provided.

INTERNET-BASED TECHNOLOGIES APPLIED IN 'EXPERT-ORIENTED' DISTRIBUTED DECISION SUPPORT SYSTEMS (DDSSs)

Decision support systems are intended to help persons to make decisions, usually in rather more complicated situations, so that an EIA can be regarded as one class of DSS.

Several Internet-based technologies have been applied for the construction of DDSSs and some of the resulting systems have been proposed for applications in water resources projects. They have been constructed for the most part by specialists in computer science, systems analysis and other such fields and follow procedures that are directed to use by 'experts' and even by 'decision-makers'. To the extent that they involve several persons in the decision-making process at all, they endeavour to do this in a collaborative way, so that the persons themselves are set within a framework of collaborative-based problem structuring and problem solving. By these means several of the points of view that are deemed by the 'experts' and 'decision-makers' to be of significance within the decision-making process can be incorporated within a certain pregiven structure. Persons involved in creating the computer output may participate within this structure as well. The notion is that, based upon the inputs of these interested and knowledgeable parties and the modellers, and through these system-building interventions of computer scientists and systems analysis, the 'experts' and 'decision-makers' can make better informed and possible equitable decisions. Van Stijn *et al.* (1994) provide a presentation of this kind of development.

These approaches have not usually been received with much enthusiasm by other persons, whether coming from the side of hydraulics, hydrology and water resources or from the side of the social sciences. The whole thrust of the movement of these professionals, such is now even enshrined in a string of position statements of the World Bank, is that the power to make decisions must be placed as far as possible in the hands of the persons who are the most directly influenced by the decision concerned, and not in the hands of individual 'decision-makers' and their 'experts'. The 'expert-oriented' paradigm proposed by the software engineers and system analysis professionals is then seen increasingly as counterproductive in this respect. In the same vein, most non-governmental organisations and some funding agencies, such as USA-Aid, no longer support projects in some countries that are directed to strengthening the power of government officials and other 'top decision-makers'. It is thus accepted that the persons directly influenced must be provided with the means to access knowledge and data having a bearing upon any proposed changes in the aquatic environment upon their own qualities of life and economic interests, as already introduced in the previous section. Because of the divergence in the differences in beliefs between persons and their situations relative to the environmental intervention concerned, they must be expected to produce a great variety of judgements, and normally no consensus can be expected. Some groups of persons will coalesce around certain beliefs, values, intentions and specific interests, whereas other groups will form around others. Some of these groups may share same beliefs, values, intentions and interests with others, but in many cases they will be in conflict. In this situation, collaboration between persons and parties will be at best limited to a sharing of facilities for the making of judgements following strings of the type shown in (1) and, in subsequent negotiation, legal and political actions, following strings of the type shown in (2).

It is axiomatic in the design of such systems – and indeed it follows from their phenomenology – that collaboration and confrontation must proceed simultaneously and the almost inevitable conflicts can only be resolved by negotiation, with socio-economic compensations between parties. The incentive for any party to use a DDSS is then to achieve a best overall position and terms of compensation, so that the DDSS follows the general rule of knowledge distribution and redistribution systems generally, that they provide means to translate changes in knowledge relations into changes in power relations. Clearly in this situation there are many persons who become stakeholders in the water resources concerned because they are empowered as stakeholders by these means. Equally clearly, these are no 'experts' and essentially no 'decision-makers' in the sense of the earlier, and conceptually pre-Internet, paradigm.

Bearing in mind, on the other hand, that the limited development of DDSSs in our field of work has for the most part continued along the lines of the earlier paradigm and the problems of DDSS analysis and design are still commonly posed in terms of this earlier approach, the way in which this approach employs current tools remains of interest. One of the challenges in this area is thus that of taking up as far as possible the methods and tools that have been developed for supporting the 'expert-oriented' line of advance and turning them around 'through 180 degrees', so to say, so that they proceed in the opposite direction in the sociotechnical sense.

In the same vein, the technical journals on Internet computing, and especially those on electronic commerce, concentrate attention on maximising certain specific institutional, or corporate, benefits, as expressed in the terms of social, or monetary, values. These are normally not at all appropriate in many of the situations involving water resources, where intrinsic values are much more normal: 'compensation values' then, for example, often bear little relation to 'replacement values' since 'replacement' is not a viable option. The enabling technology, none the less, again remains of interest.

A key architectural concept on the technical side of 'expert-oriented' decision support systems is that of *model and data integration* (distributed on the network) into an *open architecture,* giving a distributed nature to these kinds of DSS, at least in principle. An open architecture is then really only open in the very restricted, purely technical sense that every user has in principal access to the details of the construction and interpretation of the system. This of course in no way implies that the non-expert person has such access in practice: many more and quite other features than model and data integration have then to be incorporated so as to configure tools such that these non-experts can correctly use and manipulate the system. The first of the technical problems of developing such an open standard for use with hydroinformatics systems in 'expert-oriented' integrated water management applications is, however, thereby reduced simply to a problem of how to derive the appropriate vocabulary for both the *communication* and *storage requirements* of a DSS. Another problem even in this exceedingly constrained specification is that a lot of the data necessary for decision making are not quantitative data at all. The only main problem that then remains however if the problem domain is so restricted, is the integration of engineering and administrative decision information in a single DSS framework. WWW on Internet with its multimedia and communication nature and client–server architecture, then greatly simplifies (and some researchers say effectively trivialises) the communication problem in the implementation of collaborative DSSs. Based on available Internet technologies for interactive programming and remote modelling, as described in the next section, this simplification allows several viable options to be developed towards a DSS that copes with problems of communication on this narrowed field of application, and these may provide some of the means required to proceed to much more 'democratic' systems.

The most general solution then can be that communication is handled by a communication layer which can be used to link any combination of control or processing entities to any other on the network (Internet, intranet and extranet). This communication capability then becomes the basis on which the complete DSS can be built. This approach delivers distributed storage as well as distributed processing, and integrates human and machine information at the required level (Ben-Shaul & Kaiser 1998). Local processing instances at the client side are so far restricted to simulation models, knowledge-based systems and information retrieval services that are executed locally. These can be implemented as applets that can team up with instances at the server side to deal with a specific task and thus become distributed applications. Another novel approach, directed specifically to deal with the more general problem, is to develop a communication framework suitable for human as well as automated processing. Such a framework may be based on intelligent software agents (see Jonoski 1999) allowing participants in such a collaborative process to interact with intelligent agents that can exchange messages which can be used to manage the information, and then not only on the data level but also on a semantic level. Such a collaborative DSS could in principle be developed into a powerful tool in future hydroinformatics systems for solving water resources problems addressing environmental and socio-economic impact assessment in the more general 'non-expert' sense.

There is also some quite active research proceeding in the 'expert-oriented' line in the direction of developing collaborative DSS working over communication networks such as Internet. Lotov *et al.* (1996) have demonstrated the implementation of the Point Associated Trade-off technique (PAT) in collaborative decision support systems using Internet and utilising interactive decision maps. Carver (1996) also uses interactive GIS maps to show the potential use of Internet for collaborative decision making support in the field of open-land waste disposals. Several projects for collaborative DSS using Internet are currently under development. One of the examples is the so-called SDSS project (*weber.u.washington.edu/tjmoore/csdm.html*) which is a collaborative, interdisciplinary project to develop a scalable DDSS for economic and environmental planning, which can be used to quantify the economic and water quality impacts of nutrient, pesticide and sediment management strategies at the field, watershed and basin level. The SDSS is modular in design and relies on GRASS-GIS to link the economic, biophysical, water quality and hydrologic models. The SDSS uses recent developments in GIS, simulation, and Internet technologies. Another project the goal of which is to deploy a DDSS on the Web using Java technology, is the DESERT project (*www.iiasa.ac.at*). This DSS was developed as a stand-alone Windows-based application using object-oriented programming (C++) with the main purpose of providing a convenient, highly integrated tool for decision support for water quality management on a river basin scale. The US Department of Agriculture is similarly running several projects on developing collaborative DSSs the primary objectives of which are understanding the effects of agricultural management practices on water quality and developing improved management practices which are environmentally sound as well as profitable to the farmer or rancher, with an emphasis on computer networks such as Internet.

Although the philosophy of the applications envisaged in the present case is so radically different, and even opposed to most of these endeavours, the task of hydroinformatics is still to build some common ground between them. For example, to sustain any claims to be even so much as a potentially transparent system, suited to use by potentially very many non-experts, a collaborative DDSS should provide at least three basic services (see Figure 1).

Client application: This is an application, which provides the users with basic tools, such as a graphical user interface, a set-up program, client-side discussion-platform tools, etc. (For the MikeImpact tool presented below, this is a set-up program. It installs a service on the remote computer on the client side to run the tool across Internet under browser.)

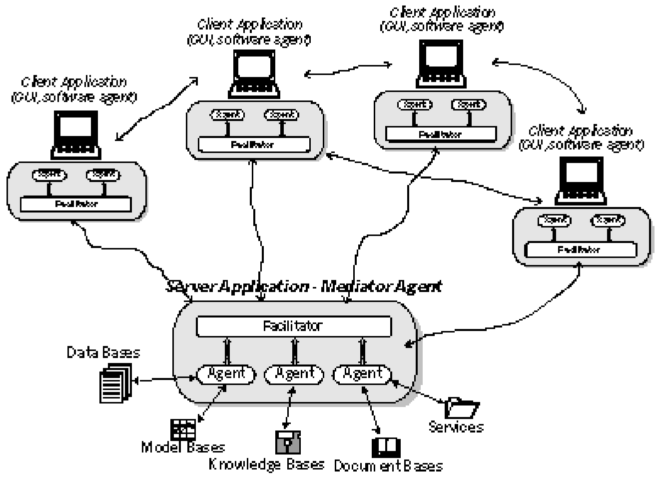

Figure 1 | Architecture of collaborative DDSS as a multi-agent construct.

Server application: This is an application built on the server side with which one can manage and broker the information about the (potentially many) client applications. It may contain the database of a specified project for which MikeImpact is provided as a means of encouraging public discussion, and which is stored on the server computer. It may also include information on states of other client applications, licenses and other such information. Furthermore, it may contain a knowledge base, a document base with case studies, articles and publications and a model base with already instantiated models (corresponding to different scenarios).

Discussion platform (or negotiation platform): This is an Internet platform for facilitating and promoting the interactions between different persons, situated on the client side, and providing interaction also between these clients and the server.

The Web-MikeImpact tool presented in this paper is a representative of the *client application* of a DDSS. It provides users with tools to run MikeImpact across Internet, and feedback is expected from the clients. It is this server application and potential discussion platform that has been constructed in such a way as to provide a basic DDSS that could later form the basis for a system that is as transparent as possible within the constraints of the methodology employed.

INTERNET TECHNOLOGIES FOR DEVELOPING DDSS

So far there are no technologies and infrastructures that address all of the requirements for building even exclusively collaborative DDSSs over the Internet. There are, however, several commercial and experimental frameworks that address some of the issues discussed in the previous section. In the commercial sector, the most prominent technologies and frameworks for collaborative and distributed computing are CORBA, DCOM and Java RMI (Velickov *et al*. 1998).

Common Object Request Broker Architecture (CORBA) is a technology that enables distributed computing across heterogeneous systems over the Internet. CORBA specifies the complete architecture necessary for communication between distributed objects. The Internet Inter-ORB Protocol (IIOP) is the most important piece of CORBA. It provides solutions for the interoperability of objects that are not tied to a specific platform of implementation, such as Microsoft's Distributed Computing Object Model (DCOM). The heart of CORBA is an Object Request Broker (ORB) which is like an object bus. The job of the ORB is to act as a sort of middle platform, allowing objects to make requests of each other. Brokering requests and returning results is the main job of the ORB: intercepting each request from one object to another, locating the object that is supposed to handle the request, invoking the appropriate method in the receiving object, passing parameters if necessary, and returning the result to the object that made the request. Because the ORB handles requests transparently, it is not important whether the request is from a local or a remote object. The strength of the ORB is that it handles these requests regardless of the programing language, operating system or platform employed, as schematized in Figure 2.

The *Distributed Component Object Model (DCOM)* serves as the base for Microsoft's component computing framework and is a part of the ActiveX technology. ActiveX is a set of core technologies that provides cross-platform, component interoperability across networks, including the Internet. In fact ActiveX is a brand name for Microsoft Component Object Model (COM). ActiveX includes COM, so as to enable communication between client components (models and modelling systems), and Distributed COM (DCOM), to integrate components (models and modelling systems) across the network (see Figure 3). ActiveX includes both, client and server, technologies (see Figure 4).

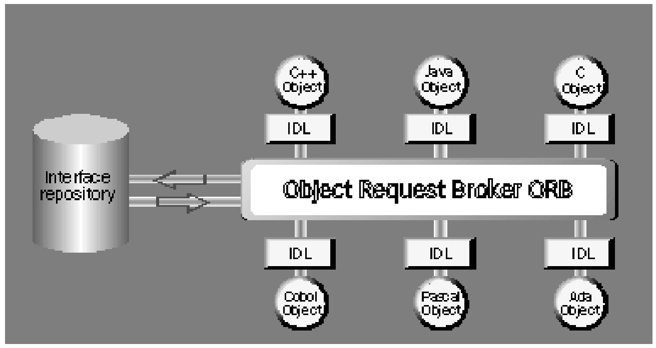

Figure 2 | A basic CORBA model, showing how the ORB mediates among objects using an interface repository (after Stanek 1997).

Since ActiveX is language independent, almost any traditional development tool can build and deploy ActiveX controls. The most popular tools include Borland's Delphi, Powersoft's PowerBuilder, and Microsoft's Visual Basic, Visual C++ and others (see Figure 5).

DCOM based on ActiveX technology provides many of the same services that CORBA does. The main advantage is its efficiency, because it is a binary standard. On the other hand, unlike CORBA, it is not platform- or operating-system independent, but only Windows based.

Java Remote Method Invocation (RMI) enables developers to create distributed Java-to-Java applications (within an homogenous environment), in which the

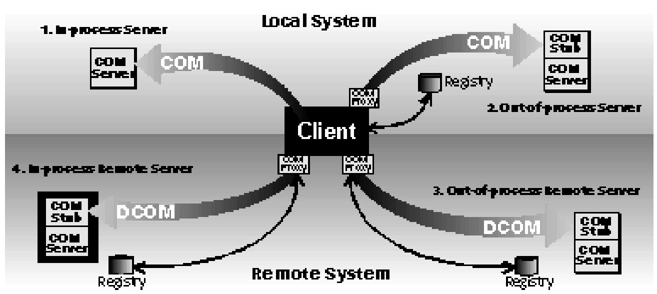

Figure 3 | The four pathways of ActiveX (after Chappel & Linthicum 1997).

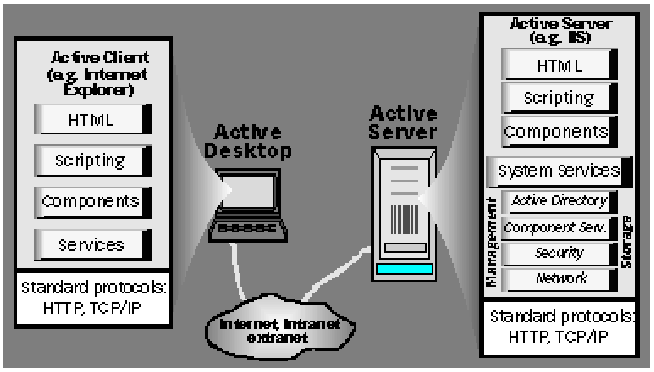

Figure 4 | ActiveX client and server technologies.

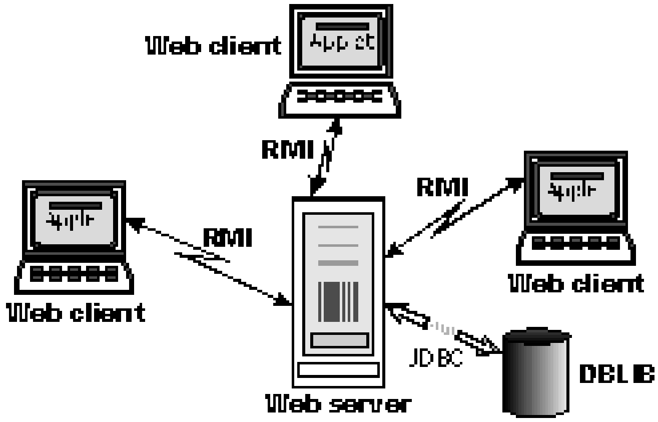

Figure 6 | Schematic representation of the basic RMI model.

methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts. Specifically, Java RMI enables developers of distributed Java applications to treat remote objects and their methods very much like normal Java objects. Java RMI brings a new level of functionality to distributed programs with features like distributed, automatic management of objects and passing objects themselves from machine to machine over the network (e.g. Internet). RMI provides a simpler implementation model for distributed computation than CORBA and ActiveX (DCOM). These objects can be new Java objects, or can be simple Java wrappers around existing objects. One of the most powerful characteristics of RMI is that it allows developers to move behaviour, such as *agents* and *business logic*, to the part of the Internet where it makes the most sense. RMI extends the Java model to be run everywhere (i.e. it is again platform independent). Figure 6 schematises the basic RMI model.

The *transparency* of a DSS can be enhanced by using any of the above described frameworks and technologies because they integrate Internet and client–server computing models. As a result, users can browse applications, very much as they browse Web pages, so as to select those descriptive devices that are the best suited to their own personal backgrounds and requirements. ActiveX technology is applied in this study for deploying the Mike-Impact judgement engine on the Internet.

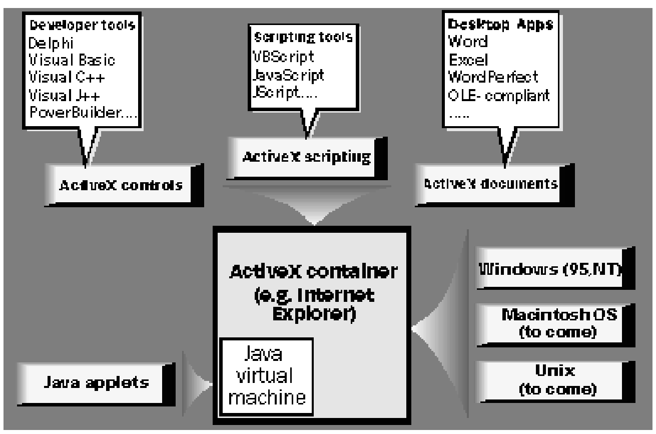

Figure 5 | Tools for ActiveX implementation.

WEB-MIKEIMPACT: A PROTOTYPE OF AN INTERNET-BASED DDSS

The Internet provides the means for a (potentially large) number of persons to make judgements in a more or less uniformly structured manner. The set of such judgemental processes comes to constitute a category correspondingly (Abbott & Dibike 1998*a*, *b*). The problem that is then posed in the first place is that of accessing a judgement engine with the Internet.

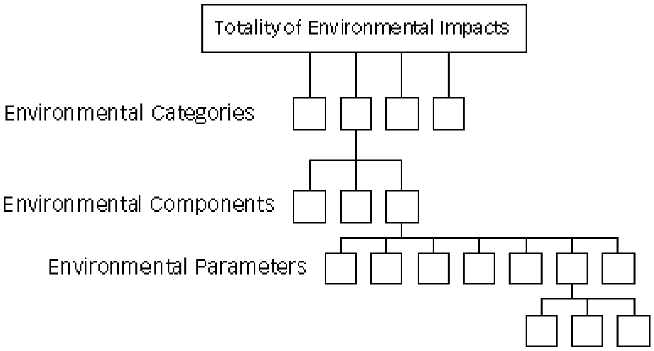

Figure 7 | Hierarchical structure of the EES.

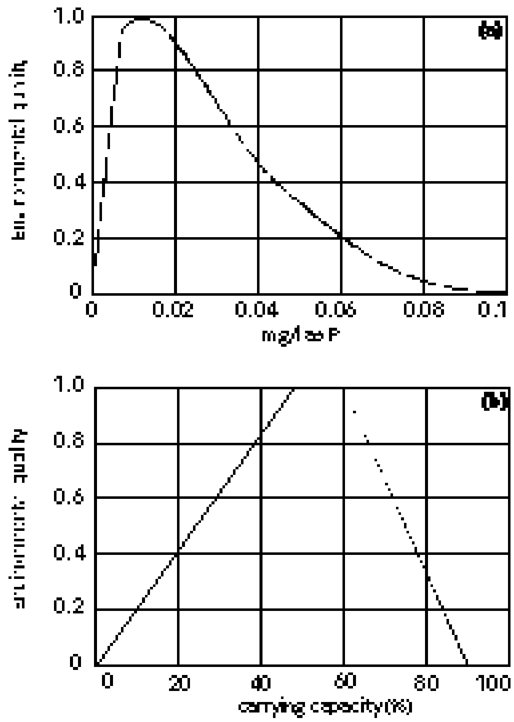

Figure 8 | Typical graphical representations of perceptions of environmental quality. (a) Value functions for inorganic PO_4; (b) value functions for browsers and grazers.

Judgement engine of MikeImpact

The judgement engine presented in this paper, the Mike-Impact system, is a standard commercially available product (e.g. DHI 1997). It is based upon an environmental evaluation system (EES) generally known as 'the Battelle method' (Dee 1973; Vis 1975). This was originally designed for use in evaluating the environmental impacts of the US Bureau of Reclamation's water resources developments. It is hierarchically structured to account for the different levels of information used in the impact analysis. Four levels of information are used:

1. Most general information: *Environmental categories*;
2. Intermediate information: *Environmental components*;
3. Specific information: *Environmental parameters*;
4. Most specific information: *Environmental measurements*;

These four levels of information are related schematically in Figure 7.

Representations of attitudes and positions in numerical and graphical forms

The value to their own persons that the users of the system place upon the occurrence of a particular numerical value of an environmental parameter, called the Environmental Quality (EQ), is expressed, on a scale between 0 (very bad) and 1 (very good), as a function of the numerical value of the parameter concerned. This is done for each and every parameter, if necessary by default. The resulting set of graphs, as exemplified in Figure 8, constitutes, in effect, a mapping from a set of possible facts, each represented by data, into a set of possible attitudes towards these facts.

This mapping is effectively realised by, and thus reflects in its turn the belief system of the individual participant in the manner expressed in Equations (1) and (2). The further mapping from attitudes to positions is realised by introducing a 'parameter importance unit', or PIU, for each parameter in turn. In the original Battelle procedure, these were decided by a standard psychological scaling technique that is usually described as a 'ranked pairwise comparison method'.

This approach proceeds through six simple steps upon the set of elements entering at each and every level in the hierarchy of Figure 7 in turn, as follows:

1. Rank the elements to be evaluated in order of their perceived importance. Thus in the intermediate level, shown in more detail in Figure 9, there are three elements under 'hydrology', namely 'low flow régime', 'flood régime' and 'change of water table'.

2. Assign a value of 1 to the first element and then compare the second element with the first so as to determine how important the second is in comparison with the first, expressed as a decimal x, $0 < x < 1$.
3. Continue this comparison by taking the third with the second element, and so on, until the end of the list.
4. Take the sum of the weights attributed within each element and divide the individual weights by this sum. It should be observed that we have to do here with *pairwise* comparisons exclusively, so that, for example, the relative weight of the third-most significant weight to the second-most significant weight is divided by this sum, without reference to the relative weight of the second-most to the first.
5. Each element in the hierarchy has a number of sub-elements, as exemplified in Figure 9 by the element 'hydrology', which has three sub-elements. The weight of each sub-element obtained from step 4 is then multiplied by the reciprocal of the number of sub-elements, the sum of the resulting weights is taken and the weights then decreased by the reciprocal of this sum. This process is intended to compensate for the greater or lesser influence of weights attributed at one level of the hierarchy upon weights at lower levels.
6. A total of 1000 parameter importance units are usually available and these are distributed downwards through the hierarchy. Thus, to take the lement 'hydrology', this may have been attributed 600 points as a result of this process applied at the top level, and if a weight of 0.5 is placed on 'low flow régime', of 0.2 on 'flood régime' and 0.3 on 'change of water table' at step 5, then PIUs of $0.5 \times 600 = 300$, $0.3 \times 600 = 180$ and $0.2 \times 600 = 120$ will be placed on these sub-elements and carried with them, to be redistributed further down the hierarchy.

In the Battelle procedure, Environmental Impact Units (EIUs) are introduced as products of PIUs and EQ for each and every parameter that is considered relevant. The judgement on the desirability of the project, and following

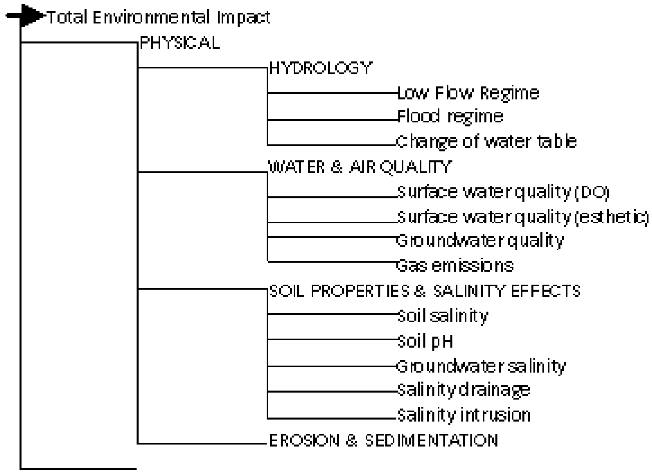

Figure 9 | Graph of sets.

this the decision to proceed with the project, is then made on the basis of the difference between the sum over all EIUs for the existing situation and the sum of all EIUs for the proposed new situation.

Referring to Equation (1), it is seen that steps 1, 2 and 3 are knowledge elicitation processes that map beliefs into attitudes, steps 4, 5 and 6 map attitudes into a position, and the last stage takes a position into a judgement.

The changes that are needed when the Battelle procedure is to be used by many non-expert persons who have a more or less major interest in a particular intervention have been enumerated elsewhere (e.g. Abbott & Jonoski 1998; Jonoski & Abbott 1998). These studies have shown that a new quality has to be introduced through the use of such systems, which is that of *transparency* during *co-operation*. Experience with an earlier internetted system, the EAGLE, that has been operational from 1995 onwards showed, in effect, that the implication strings of one user of the system, as exemplified by (1), had to be made as transparent as possible to other users if a co-operative attitude towards the overall environmental impact assessment was to be maintained between the various users. This, however, necessitates the support of a process of the kind schematised in (2), and indeed 'transparency' can be defined as the property whereby the process symbolised by (2) is made explicit to all users of

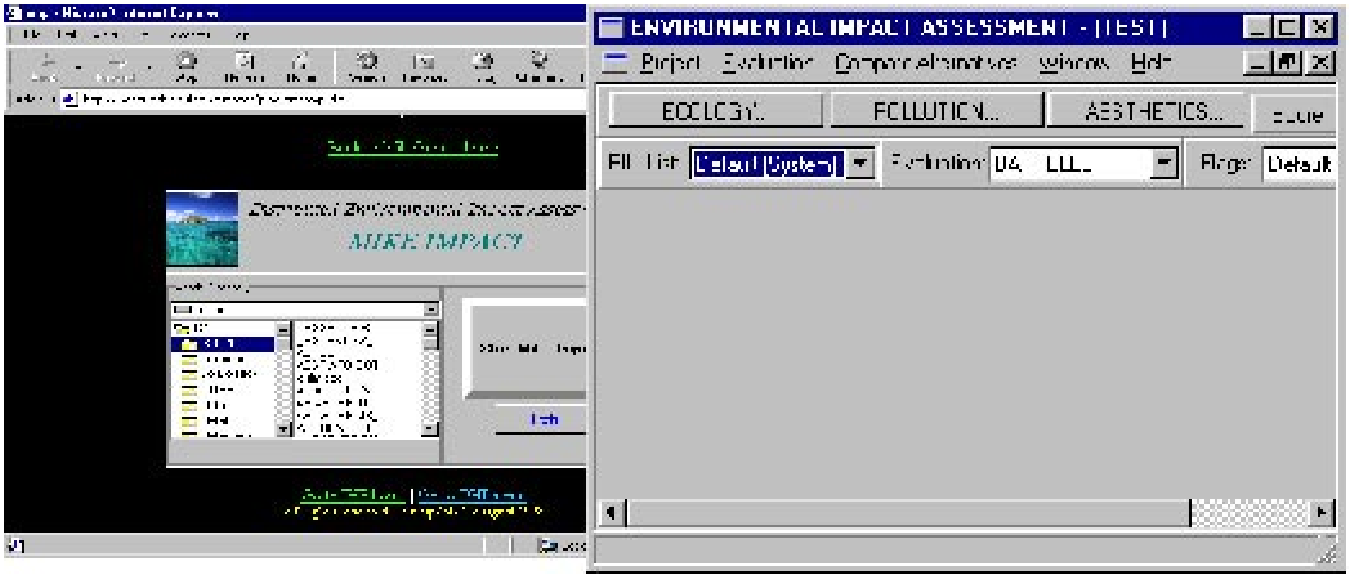

Figure 10 | ActiveX control of Web-MikeImpact.

the system. Thus, by working backwards through the processes 6, 5, . . . , 1 above, as provided by a user of the system, it may be possible to arrive back to the belief system of that user. The commercial system used in this study was itself based upon two working prototypes, called *Cascade1* and *Cascade2*, that provided marked increases in transparency (Shipton, née Simic, 1993). It is clear, none the less, that existing approaches leave much to be desired in this direction. Moreover, transparency is essentially a relation between the states of minds of individuals, so that a 'transparent system' can never be more than one that strengthens this *mental* relation.

Deployment of Web-MikeImpact on Internet

The MikeImpact judgement engine was originally developed by DHI (DHI 1997) as a stand-alone Windows-based application. Using the source code of MikeImpact, all the original forms and components were transferred into ActiveX components and thereafter embedded into an ActiveX container, such as a Web browser (see Figure 10).

In order to set-up the client's environment (database connections and connection with the host server) for running Web-MikeImpact within the browser, an initialisation program was developed which has to be installed on the remote computer in advance. Users can download the installation program from the Web-MikeImpact home page or run active set-up remotely over the Internet.

For the non-experienced users of the MikeImpact judgement engine, several on-line tutorials were developed demonstrating how to start and create a new project, carry out parameter selection, evaluation, analysis/score, etc. All the demonstration tutorials can also be accessed via the Web-MikeImpact home page available on *http://www.hi.ihe.nl/hi/test/mikeimpact/mikeindex1.htm*. The home page also provides facilities for submitting the EIA user's feedback, which is processed and analysed on the server.

CONCLUSIONS

An overview of the Internet technologies applicable for building DDSSs has been given, and one component of an Internet-distributed environmental impact assessment toolset has been constructed by way of a feasibility study. The basic tool was a *MikeImpact* system, provided by DHI, so that the resulting Internet-distributed tool became a *Web-MikeImpact*. Because MikeImpact is primarily a device for facilitating the making of judgements, it belongs to a class of tools called *judgement engines*. Such engines can be used in several ways when run over the Internet. The Internet application selected for this study was one linking a number of client machines with a central server. This application allows (potentially very) many persons to employ the same procedure in making their individual assessments. In principle, and with some further modifications, it should also allow each user to interrogate the judgement processes of other users. Judgement engines should of course be used together with the products of fact engines, as described in the introduction to this issue of the *Journal of Hydroinformatics*.

The *Web-MikeImpact* is intended to be employed already by many non-expert persons, although it will clearly need to be much modified again for applications in many societies. As such it should be employed in two very different ways and following two very different philosophies to those propagated in most such methodologies that have so far been addressed, which are 'expert-oriented' in their underlying philosophy. Although these two philosophies are so opposed, the 'expert-oriented' one not only has certain areas of application – and is almost sure to be more popular within many corporate and government bodies – but it also provides experience in the use of many enabling technologies. The greatest problems of the 'non-expert-oriented' systems in the future are, however, to be expected on their 'fact' side in the physical world and on their 'judgemental' side in the world of human societies, and much less on the enabling technology side.

This paper also serves as an experiment in the presentation of mental processes and computational/logical operations that are difficult to express using conventional natural-language, symbolic and graphical means exclusively. To come to presence, or to come alive in the mind of the reader/user, this work should be read 'alongside', or 'together with', the operation of the tool that it itself describes. This paper thus presents one possible solution to the problem of 'writing' (in the specific Saussurian sense of *écriture*) about processes proceeding interactively within the minds of humans and in electronic media. (See, originally, Saussure 1916, pp. 92–101/1993 pp. 44–54.)

ACKNOWLEDGEMENTS

The authors thank A. W. Minns at IHE for his help and advice in this study and the DHI, represented by V. Babovic, for making the MikeImpact code available for this study.

LIST OF ABBREVIATIONS

BDE: Borland Database Engine
CORBA: Common Object Request Broker Architecture
COM: Component Object Model
DCOM: Distributed Component Object Model
DDSS: Distributed Decision Support System
DNA: Distributed Application Architecture
EIA: Environmental Impact Assessment
JVM: Java Virtual Machine
MDI: Multiple Document Interface
OLE: Object Linking and Embedding
RMI: Remote Method Invocation
WWW: World-Wide Web

REFERENCES

Abbott, M. B. & Jonoski, A. 1998 Promoting collaborative decision-making through electronic networking, In *Hydroinformatics '98* (ed. Babovic, V. and Larsen, L. C), pp. 911–918. Balkema, Rotterdam.

Abbott, M. B. & Dibike, Y. B. 1998*a* On the representation of processes in the postsymbolic era. In *Hydroinformatics '98* (ed. Babovic, V. and Larsen, L. C), pp. 1177–1184. Balkema, Rotterdam.

Abbott, M. B. & Dibike, Y. B. 1998*b* The symbolic representation of hydroinformatics processes using elements of category theory. In *Hydroinformatics '98* (ed. Babovic, V. and Larsen, L. C.), pp. 1185–1192. Balkema, Rotterdam.

Ben-Shaul, I. & Kaiser, G. 1998 Coordinating distributed components over the Internet. *IEEE Internet Computing Journal* **2** (2), 83–86.

Brentano, F. 1862/1960 *Von der Mannigfachen Bedeutung des Seienden nach Aristoteles*. Herder, Freiburg in Breisgau/Olms, Hildersheim.

Chappel, D. & Linthicum, D. S. 1997 ActiveX demystified. *Byte Magazine*, September: *http//www.byte.com*.

Carver, S. 1996 *Example of DSS for Waste Disposal Using CGI and GIS Maps*. *http://www.ccg.leeds.ac uk/mce/mce-home/htm*.

Dee, N. *et al.* 1973 *Final Report on Environmental Evaluation System for Water Resources Planning.* Bureau of Reclamation, U.S. Department of the Interior, Battelle Columbus Laboratories.

Husserl, E. 1938//1957 *Experience and Judgement, Investigations in a Geneology of Logic* (trans. Churchill, J. S. & Ameriks, K.). Routledge, London.

Jonoski, A. & Abbott, M. B. 1998 Network distributed decision-support systems as multi-agent constructs. In *Hydroinformatics '98* (ed. Babovic, V. and Larsen, L.C.), pp. 1219–1226. Balkema, Rotterdam.

Jonoski, A. 1999 Towards The Development Of Distributed Decision Support Tools For Use Over Electronic Networks, in-press for *XXVIII IAHR congress*, 22–27 August, Graz, Austria.

Lotov, A. V., Bushenkov, V. A. & Chernov, A. V. 1996 *Experimental Internet Resource for Development of Independent Strategies*. *http://www.ccas.ru/mmes/mmeda/*.

Meinong, A. 1913 *Gesammelten Abhandlungen, Zweiter Band: Abhandlungen zur Erkentnistheorie*. Barth, Leipzig.

MikeImpact User Guide 1998 IHE lecture notes, HH370/98/1.

Saussure, F. de 1916/1993 *Cours de Linguistique Générale*. Payot, Paris.

Simic, S. 1993 A Decision Support System for Environmental Assessment, M.Sc. Thesis, HH367, IHE, Delft.

Stanek, W. R. 1997 Distributed programming with CORBA and IIOP. *PC Magazine* **16** (16), September.

Van Stijn, T. L., Blokland, P. A. & Lander, J. 1994 Integrating systems for the challenge of coastal zone management. In *Proceedings of the First International Conference on Hydroinformatics* (ed. Verwey, A., Babovic, V. B., Minns, A. W. & Maksimovic, C.). Balkema, Rotterdam, The Netherlands.

Velickov, S., Price, R. K. & Solomatine, D. P. 1998 Using Internet for hydroinformatics – practical examples of client/server modelling. In *Hydroinformatics '98* (ed. Babovic, V. and Larsen, L. C.), pp. 965–972. Balkema, Rotterdam.

Vis, M. 1995 *Environmental Impact Assessment*, IHE lecture notes, HH324/95/1.

World-Wide Web

Internet Solutions; Internet, Web Solutions; E-Commerce; Java programming, *http://www.termondt.com/interactiv*. #Intranet and Extranet Development.

Windows DNA-Related Articles, *http://www.microsoft.com/dna*

ActiveX Controls, *http://www.microsoft.com/com/activex.htm#books*

ActiveX Magic: An ActiveX Control and DCOM Sample Using ATL, *http://www.microsoft.com/com/overview.htm*

Building ActiveX Controls with Delphi 3 *http://pw2.netcom.com/~cherrman/d1030.htm*

Delphi 3 – The Active Foundry, *http://informant.com/delphi/dt9705.htm*

Web-MikeImpact test site, *http://www.hi.ihe.nl/hi/test/mikeimpact/mikeindex1.htm*

Experimental site of Web technologies in hydroinformatics, *http://www.ihe.nl/hi/test*