Chapter 6
Elbe DSS: A Planning Support System for Strategic River Basin Planning

Bernhard M. Hahn, Sebastian Kofalk, Jean-luc De Kok, Jürgen Berlekamp and Mariele Evers

6.1 Introduction

The German part of the Elbe River and its basin are characterized by multiple problems and objectives that call for strategic management based on an integrated approach. In August 2002, the region suffered a catastrophic flood with loss of lives and damage amounting to over 9 billion Euro. During the summer months, because of low flows, shipping along the river is problematic, which considerably reduces the economic transport capacity of the river. Several areas along the river act as a habitat for rare plant and animal species and have been designated as nature reserves. The output of diffuse and point sources of pollution in the river basin must be controlled in order to comply with standards of the *EU Water Framework Directive* (EU 2000).

Management actions taken as a part of implementing policy interact at different spatial and temporal scale levels, which complicate the identification of promising river management strategies. For example, the restoration of poorly maintained groynes to improve the navigation conditions may lead to undesirable changes in the hydrodynamic character and ecological vegetation patterns of the floodplains. Furthermore, long-term trends in socio-economic development and climate change can be expected to interact with measures. The diversity and structure of river-related research and planning organizations in Germany, the fact that the Elbe River cuts across six different federal states and that one third of the basin belongs to the Czech Republic, also complicate the formulation of promising river management strategies.

To deal with these problems, the German Federal Institute of Hydrology (Bundesanstalt für Gewässerkunde or BfG) took the initiative for a three-year project to develop the *Elbe DSS*, a pilot version of a planning support system (PSS) for strategic integrated management of the Elbe River and its basin. The *Elbe DSS*

B. M. Hahn (✉)
Research Institute of Knowledge Systems (RIKS)
PO Box 463,6200 AL Maastricht, The Netherlands
E-mail: bmhahn@riks.nl

S. Geertman, J. Stillwell (eds.), *Planning Support Systems Best Practice and New Methods*,
DOI 10.1007/ 978-1-4020-8952-7_6,

was completed in 2006 and includes measures and indicators for flood safety, water quality, navigation and vegetation ecology. The user can select from several demographic and climate scenarios and apply various combinations of management measures to explore their effects at different spatial scales ranging from the river basin to specific locations of interest along the river. Each measure can be configured interactively using a specialized graphical user interface that presents the underlying model parameters at an aggregation level appropriate for strategic decision making. All geo-information is accessible in a built-in map viewer/editor, which can also be used for interactive configuration of measures by general users.

In view of the problems with respect to the adoption of PSS that have been mentioned by Walker (2002) and Westmacott (2001), stakeholders and potential users were involved in the design of the system from the beginning. Potential user organizations, key problems and tentative measures were identified during a feasibility assessment that preceded the project. Scientific or so-called ‘research’ models are often data and location dependent and can be computationally demanding. This makes these models less suitable for application in an interactive instrument to be used during sessions with stakeholders where the instrument should be flexible enough to represent different management options or new political priorities as they emerge during discussion. Therefore, the *Elbe DSS* is based as much as possible on simple models, or simplifications of models that were too comprehensive for direct incorporation into the tool. Implementation of the DSS within organizations involved in river-basin management is currently being coordinated by the BfG through a series of testing and training workshops with users. The outcomes will be used to further refine the design in order to meet the expectations of the practitioners.

In Section 6.2, we provide a broad definition of *integrated special decision support systems* (ISDSS) as a specific type of PSS and we define how they can be used within the context of strategic river basin planning. A brief overview of the design methodology adopted for the development of the *Elbe DSS* is given in Section 6.3. In Section 6.4, we describe the key roles and interactions in a typical ISDSS project team. Knowledge integration is the main challenge of ISDSS development. We discuss our approach to knowledge integration in Section 6.5. In Section 6.6, we give an overview of the techniques used for technical model integration. In Section 6.7, we describe some aspects of the realization of the *Elbe DSS* prototype, and in Section 6.8 we report preliminary results of an evaluation of the prototype by users. Finally, in Section 6.9, we summarize our conclusions from the development of the *Elbe DSS* and indicate possible directions for further research.

6.2 DSS for River Basin Management: State of the Art

There is no generally accepted definition of what a decision support system (DSS) is. Table 6.1 shows which of the common DSS characteristics identified in Marakas (1998) are present in the *Elbe DSS*.

Table 6.1 Common DSS characteristics

Common DSS characteristic	*Elbe DSS*
Employed in semi-structured decision contexts	Yes
Intended to support rather than replace decision makers	Yes
Support for all phases of the decision-making process – for a model of the decision-making process, see Mintzberg (1976)	Yes
Focus on effectiveness of the decision-making process rather than efficiency	Yes
Is under control of the DSS user	Yes
Uses underlying data and models	Yes
Facilitates learning on the part of the decision maker	Yes
Is interactive and user friendly	Yes
Developed using an evolutionary, iterative process	Yes
Provides support for all levels of management	No
Can provide support for multiple independent or interdependent decisions	Yes
Provides support for individual, group and team-based decision-making contexts	Partially

Source: adapted from Marakas (1998).

Harris (1989) introduced the term planning support system (PSS) and defined it as an architecture for coupling a range of computer-based methods and models into an integrated system for supporting planning functions. DSS can be seen as part of PSS, when the scope of the DSS is more on the operational level. However, in the context of our view of strategic river-basin planning, we use the term PSS for a special type of DSS with distinguishing features as described in Table 6.2. PSS and policy support systems (PoSS) are developed to support the long-term, strategic planning process as opposed to management support systems (MSS), which are intended to support short-term decision making. An overview over the recent history and classification of PSS is given in Batty (2007) and policy support systems are defined in Kok (2007).

Table 6.2 Features of PSS as a subclass of DSS

Decision Support Systems (DSS)	
Management oriented DSS	Planning Support Systems (PSS) Policy Support Systems (PoSS)
Management oriented	Policy and planning oriented
Short-term/immediate	Long-term/strategic
Concrete topics (complexity due to technical limitations of the systems and methods used)	Abstract topics (complexity due to lack of knowledge and consensus on objectives)
Specialised sectoral context	Broad societal context
System based on GIS, database manipulations and (few) sectoral models	System based on complex integrated model(s)
Optimization oriented	Simulation and exploration oriented

Strategic river-basin planning may be considered as the attempt to control a complex and dynamic system of interdependent natural and human-driven processes. Due to these characteristics and the fact that experimenting with a real world system is not an option, information systems designed to support environmental planning are often model-centric. Model-centric PSS represent knowledge about the domain in formal models which can be used for analysis or simulation. Borshchev (2004, p. 1) describes a simulation model as *a set of rules (e.g. equations, flowcharts, state machines, cellular automata) that define how the system being modelled will change in the future, given its present state*. To accurately reflect the connectedness and interactions of the real world system, model-centric PSS typically couple several domain-specific models in an integrated system model (ISM). Each of the individual models then represents a process that is considered relevant within the planning and decision context. The system model is integrated, because it explicitly represents and simulates the interactions between the processes. Since, in the domain of environmental planning, most processes and variables of the ISM have a spatial dimension, we use the term *integrated spatial decision support system* (ISDSS) for a spatial PSS with an ISM as its core component. The basic assumption of the simulation approach is that the ISM mimics the behaviour of selected, relevant aspects of the real world system. Therefore, running simulations with the ISM enables the decision maker to explore a space of possible strategies prior to taking action.

The generic usage context of an ISDSS is shown in Fig. 6.1. Physical, ecological and socioeconomic processes are represented in the ISM as a network of coupled processes. The ISM embodies our current knowledge about the system at a level of abstraction that is appropriate for policy design and strategic planning. Apart from its autonomous behaviour, the state of the system is influenced by: (i) policy options; and (ii) external influences. Policy options represent variables of the system which are under human control and therefore are related to planning and decision making.

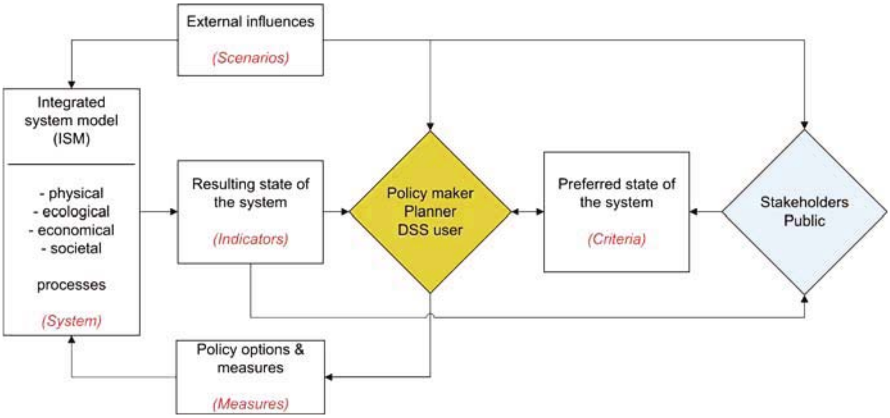

Fig. 6.1 Generic usage context of an ISDSS

External influences represent exogenous factors such as climate change. These influence the system, but are not under direct control of the decision maker, nor do these factors receive direct feedback from the system. In principle, the ISDSS user has access to all output variables of all processes represented in the ISM. However, policy formulation requires indicators that represent policy relevant aggregated information about the system. The relevance of such indicators depends on the policy context in which the system operates, and should be widely accepted among scientists and stakeholders. Indicator values can be analysed with respect to various criteria. Experimentation and exploration by hand or algorithmic optimization routines can be used to adapt the parameters of multiple combinations of measures in order to reach a close match between the resulting state of the system (indicators) and the preferred state of the system (criteria).

6.3 Design Methodology

The development of an ISDSS involves multiple disciplines and multiple roles and is an iterative process where the product is realized by interdisciplinary collaboration between scientists, modellers, domain specialists, IT specialists and end users (Fig. 6.2).

The aim is to develop a balanced product as represented by the position of the arrowhead in the centre of Fig. 6.2. To achieve this goal, ISDSS development ideally starts in the policy part of the triangle. Policy makers and planners should be involved in the project as equal partners from the start, or even better, be the initiators of the project. This increases the chance that the design of the system is driven by real world policy and planning requirements. Numerous projects have been initiated as part of a multi-disciplinary research project where the main motive for ISDSS development was to integrate the sectoral scientific results of other (sub)projects. In

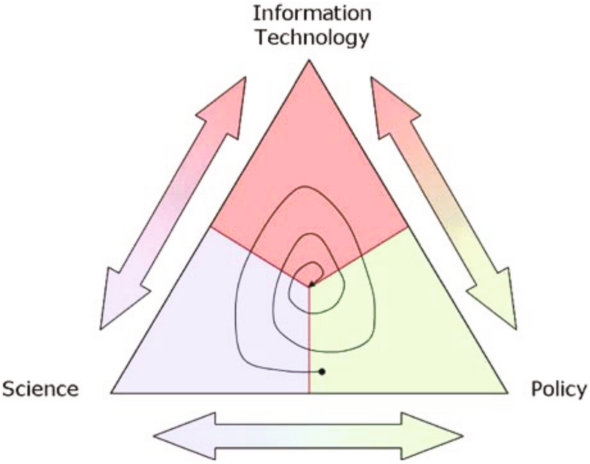

Fig. 6.2 Iterative ISDSS development for policy and planning support

general, projects set up in this way tend to serve the (legitimate) academic interests of the participating scientists more than addressing real policy and planning issues. Among the characteristics of the ISDSS products that come out of these so-called science-driven projects we find that:

- the ISDSS is based on weakly integrated comprehensive scientific models;
- models often do not reflect the scope of policy design questions: very detailed answers for questions in a very restricted context instead of sketchy answers in a broad context;
- complex user interfaces that expose the complexity of the underlying models to the extent possible;
- sluggish performance and weak technical integration due to insufficient software engineering quality of the individual models; and
- weak integration in the organisational context and existing IT infrastructure of the end user.

GIS-based DSS tend to be tools that analyse the past and the present but often lack the underlying integrated models necessary to explore how a complex system might evolve in the future. Therefore they tend to be more management oriented DSS that deal with operational and/or short-term tactical decision making rather than policy and planning oriented DSS, that deal with long-term strategic decision making.

Based on the lessons learnt from the *Elbe DSS* and other ISDSS development projects, we present here some elements of a design methodology for ISDSS which tries to avoid some of the problems discussed above with the aim of increasing the chance of developing a product that is perceived as relevant and useful for application to real world planning and policy problems. We will discuss three such elements of an improved ISDSS design methodology – project team operation, the integration of scientific knowledge and technical (software) integration.

6.4 The ISDSS Project Team

6.4.1 Roles in the Project Team

Successful development of an ISDSS is an interdisciplinary team effort. To achieve a balanced and useful product, the following key roles and responsibilities should be present in the project team.

End users provide the policy and usage context of the ISDSS. They define problems, delineate the business scope, and identify functional and organizational requirements for the system. End user representatives in the project team should consider themselves the ‘problem owners’. They should have a clear role and set of tasks in the project team and should be involved from the start. In an ideal case, the ISDSS project would be initiated and sponsored by an end user organization.

Scientists provide the knowledge about natural and human-driven processes and their interactions. They define the spatial and temporal resolution, the scale and levels of detail of the individual models and the ISM. In addition to their deep domain specific knowledge, they should have a strong motivation for the integrated and interdisciplinary approach required for ISDSS development, without compromising the scientific integrity of their contributions.

IT specialists are responsible for system architecture, technical model integration, software design and implementation. They should have in-depth knowledge of state of the art software architectures and middleware technologies and a good vision on an appropriate technical design for the ISDSS. They should be highly flexible with respect to the formats, programming languages and environments they have to work with, and be able to manage a high level of complexity in a dynamic and iterative development process.

The *DSS architect* has primary responsibility for integration, communication and management of the development process. This person takes responsibility for integrating the work of the scientists and for controlling the quality of the ISM. As a generalist, this role bridges methodological and knowledge gaps between end users, scientists and IT specialists. In order to fulfil this role, the architect should possess a very good intuitive understanding of the application domain, the (policy and planning) problems posed, along with good communication skills.

6.4.2 Interactions in the Project Team

In the context of ISDSS development, the *interaction between scientists and end users* primarily deals with the issue of selecting policy relevant research findings, models and data. Developing solutions that fit the needs of policy makers and planners requires the scientists to shift their focus from process towards problem orientation – which processes need to be included and integrated for tackling particular problems? The processes should be represented at a level of detail and at spatial and temporal resolutions that are appropriate for the policy and planning problems at hand. For adequate representation, the integrated models often require that processes are linked across various spatial and temporal scales. This can mean that existing scientific models have to be adapted or reformulated.

Relying on (the predictions of) a computer based tool as a guide for policy formulation and planning requires wide acceptance and trust in the underlying integrated model and its individual components among the stakeholders. To achieve this, instead of being a black box, the ISM should be as transparent as possible. In the *Elbe DSS*, the documentation of the individual models as well as references to papers, calibration and validation reports are accessible to the user as part of a context sensitive online help system. Calibration, validation and uncertainty analysis are notoriously difficult and tedious for coupled (system) models. Therefore, it is important that scientists and end users develop a shared vision of how to tackle these tasks during the development phase. Providing analytical and visualization tools for uncertainty analysis can help to communicate the chosen approach.

In general, the *interaction between scientists and IT specialists* mainly focuses on the technical and, to a lesser extent, the scientific aspects of model integration. The goal of technical model integration is to develop a flexible ISDSS that allows upgrading each of its components, to keep it up to date with new research findings as well the changing needs of the policy context. The state-of-the-art approach to achieve this is to use an object-oriented software framework to construct the ISM from elementary building blocks, which pertain to a level of detail that is sufficiently high to be applied by scientists and domain experts. Most ISDSS development projects use existing models rather than developing new models from scratch. Existing models often have to be technically adapted in order to fit into the model integration framework. This often gives rise to a maintenance problem, because one might end up with the original (research) version and a separate 'DSS version' of a model. Keeping both versions in a synchronized state of development can be costly. The best way to avoid this is to make sure that scientists and software engineers collaborate at an early stage in the model development process.

The aim of the *interaction between IT specialists and end users* is to develop a user-friendly system that addresses the user's questions and seamlessly integrates in her/his existing IT infrastructure and organizational context. Here, user friendliness means that the system provides an ergonomic, transparent, interactive and responsive software environment, which enables the user to explore planning alternatives without being fully exposed to the complexity of the underlying ISM. Seamless integration in the users' existing IT infrastructure and organizational context is a key success factor for acceptance and adoption of the system. The preferred approach to integration on the technical level is using open standards and interface specifications (for example, the OpenGIS® standards and specifications developed by the Open Geospatial Consortium (OGC) and available at http://www.opengeospatial.org).

Embedding the system in the organizational context of the user is far more difficult. In the first place, this requires a thorough analysis of the user's organisations' work flows and business processes; an essential part of the requirements analysis phase of any ISDSS project. Furthermore, the interdisciplinary and integrated nature of the ISDSS may require that the task and business processes of the user's organisation change to some extent. Our tools shape our thoughts – our thoughts shape our tools. In the early phases of development, users often find it difficult to specify functional requirements for the ISDSS; once they have a first prototype, it will trigger their imagination about how it should be further developed in order to provide real added value for accomplishing their tasks. Involving end-user representatives during the entire development of the ISDSS, preferably as full project members, will enhance their feeling of ownership and control of the product and the changes it might inflict on their work processes. Especially in research driven ISDSS projects, the value of frequently interacting with and understanding the end users' changing organizational context can hardly be overestimated.

6.5 Knowledge Integration

Designing and implementing the ISM requires integrating knowledge, models and data from multiple disciplines. Since standardized methodologies for knowledge integration do not yet exist, this is often the most challenging element of ISDSS development. The structure of the ISM to a large extent determines the ISDSS functionality, e.g. which scenarios, policy options and indicators can be implemented? Knowledge integration, therefore, is an essential activity of the conceptual design phase of the ISDSS development process. Knowledge integration involves some scientific, technical and end-user related aspects of which we will give an overview here.

6.5.1 Scientific Aspects

How can knowledge from different disciplines be integrated in an ISDSS and how can we link models with different scientific paradigms? The starting point of model integration should be a detailed systems analysis based on user requirements, which identifies all relevant elements and cause-effect relationships that should be reflected by the integrated systems model. What are the model selection criteria and how can we then integrate these models?

A precondition for coupling models is that their interfaces match semantically and structurally. The spatial and temporal resolution and extent of the models must correspond, or appropriate aggregating and/or disaggregating transformations must be applied. In the case of the *Elbe DSS*, the spatial resolution ranges from the whole catchment (approximately 100,000 sqkm) to river sections of 2–3 km length, whereas the temporal resolution varies from decades to daily time steps. In addition, the modelling paradigm can also vary, because processes in the real world can be represented using different modelling approaches. In the natural system as an example, the runoff and infiltration of rainfall can be described using (partial) differential equations, simpler water budget approaches or stochastic approaches. Overall, the level of detail concerning spatial and temporal scale as well as the modelling paradigm should be selected based on adequacy for and appropriateness to the decision problem at hand. Applying the principle of Ockham's razor by choosing the simplest model that can describe a particular process of interest at the appropriate spatial and temporal resolution is often a good DSS design rationale. However, simple models are not always flexible enough and our DSS requirements are sometimes conflicting. So we often need to make a trade-off between flexibility and performance.

Although models can be linked technically as black boxes (see below), this is not recommended from a scientific point of view. To ensure robust and reliable behaviour of the integrated system, it is necessary to analyse model structure and information and data flows. Analysis of those model characteristics makes it possible to define adequate interfaces and ensure structural correctness of the integrated model. In some cases, it may become clear from model analysis that two models to be integrated share the same processes. In this case, it will be necessary to modify one or both models to ensure overall model consistency. Model selection also depends on input data requirements and run-time characteristics and can be supported by sensitivity and uncertainty analyses of the coupled models and DSS system as a whole. Models with excessive data needs and run-times longer than a few minutes are less appropriate for integration in an interactive PSS.

6.5.2 Technical Aspects

How can scientific models be technically integrated, without re-writing all software code, with maximum performance, user-friendliness and at low cost? Let us consider some general requirements for technical integration of models, most of which are also applicable to the ISM and ISDSS as a whole.

Flexibility is a key aspect of an instrument for strategic river-basin planning: an ISDSS is expected to have a long lifecycle. During its lifecycle, the requirements for the system will almost certainly change, data will need updating, models will evolve and scientific knowledge will change or advance. Therefore, the integrated model, as well as the system as a whole, should be as flexible, modular and as decomposable as possible. Individual models should remain accessible as self-contained software units, preferably at the binary level, but at least at the source code level.

Maintainability: The responsibility for maintenance and the potential further development of a model should remain with the original data/model provider and normally should not be transferred to the technical model integrator. It should always be possible to independently test a model before, and after it has been linked to other models in the ISM.

Extensibility: It should be possible to extend the ISM. Individual sub-models should communicate solely via interfaces in order to be exchangeable with as little effect as possible on other parts of the ISM.

Reusability: The development of an ISM may yield potentially reusable artefacts at various levels of abstraction including: (i) the conceptual system model; (ii) the integrated system model as a whole or parts of it; (iii) individual scientific models; (iv) source code; (v) compiled code; and (vi) user interfaces. The largest gain from reuse is obtained in the earliest stages of development cycle. Reusability is almost always formulated as a key requirement of any system development that consumes substantial resources. However, a generic and reusable design always takes considerable more effort compared to a design that only works in a narrow context (Hahn and Engelen 2000). This fact should be reflected in one way or another in the business model for the system. If the system is to be reused in several regions, the extra cost for providing this flexibility and generality (see below) should not be carried by the first client alone.

Generality: If fed with the appropriate data, ideally it should be possible to apply all models to other river basins or river networks. This is also a requirement for the ISDSS as a whole. Of course, the ISM can only be as generic as its least generic part. It is therefore important to avoid decreasing the generality of the ISM by integrating models with very specific locally bound data requirements if more generic models are available.

Accessibility: Often an ISDSS is designed to assist planning and decision-making processes for multiple users with different skill levels and backgrounds. The system design may reflect this by providing alternative user interfaces for different types of users, e.g. a system oriented user interface for the scientist/expert, or a task oriented user interface for practitioners. Thinking of participatory approaches to planning, it may be important to have location independent access to the system.

Performance: Since the primary use of an ISDSS is interactive exploration of policy designs and planning scenarios, the performance of the integrated model in terms of computing speed is a critical success factor. When interactively simulating a policy measure, the update of on-screen graphs, tables and dynamic maps ideally should not take much longer than a few seconds.

The main challenges of technical model integration are to meet the above mentioned requirements, to ensure software engineering quality for the ISM, whilst not imposing too much technical complexity or rigidity on the scientists and modellers participating in development. Knowledge, models and data which are selected to be integrated in an ISDSS, are expected to have a high level of scientific maturity. The ISM, therefore, typically integrates results from previous research projects rather than newly developed models. Integrating existing models and data requires the technical model integrator to deal with: (i) various programming languages; (ii) various compilers and interpreters; (iii) various run-time environments; (iv) various operating systems; (v) various concepts for data and memory management; (vi) open and closed systems; and (vii) a plethora of *ad hoc* data formats. If at all, the technical model integrator has only very limited control on the technical design and the form in which models will be delivered to the project. Geospatial and other data may be provided to the ISDSS project in a great variety proprietary and model specific *ad hoc* formats. The technical model integrator, together with the ISDSS architect, should select and/or define a small number of standard data formats for the project. Ideally, these formats should be open, widely used and precisely documented. Binary formats usually are more memory efficient and provide better read/write performance compared to ASCII based formats. However, they are almost impossible to access without exact documentation and specialized applications. When it cannot be avoided to develop a project or model-specific data format, this should be done on the basis of open and widely used meta-formats like XML.

Typical problems technical model integrators and DSS architects encounter when integrating existing scientific model code in an ISDSS include the following.

Architectural mismatch: Many scientific models are implemented in the 'read-compute-write' FORTRAN style of the 1970s, meaning they read their input data from a file and, after doing their computations, write the results to a file. Integrating such code in a modern event driven interactive application requires non-trivial changes in the control structures and application logic of the model.

Software interface: Software implementations of scientific models are usually designed as (stand alone) research tools. They are not designed as software components that can function as part of a larger system and they seldom feature a software interface that allows their execution to be controlled by another software component such as a simulation controller. To integrate such model code into the ISM, it needs to be 'wrapped' by an extra layer of software that implements the required interface. The worst case with respect to model integration is models that are developed within a closed modelling environment and can only be executed inside that environment. In such cases the only way to integrate the model is to re-implement it in an accessible environment.

Software engineering quality: In contrast to the quality of the scientific content, the software engineering quality of scientific model code is sometimes poor. When such code has to be integrated in an ISDSS, this can create performance, maintenance or usability related problems.

6.5.3 User Aspects

The user aspects of knowledge integration add the dimension of problem orientation to the scientific goal of improving our understanding of a complex system. We want to use our improved understanding of the system to inform management action, for example to better implement sustainability as framed in the EC Water Framework Directive (EU 2000). Therefore, knowledge integration in the context of DSS development should always start from policy, planning and management questions.

To find out more about these questions, and how a DSS could effectively help to solve them in a practical and adequate way, one of the authors conducted a broad evaluation with almost 200 experts in river basin management in five North Sea riparian countries (Evers, forthcoming). The evaluation confirmed that generally DSS are seen as a feasible means of supporting *integrated river-basin management* (IRBM). However, when searching for DSS requirements, one main issue is to find the tangible added value of a system – why and for what purpose is it worth investing financial and human resources into the development of a DSS for river-basin management? Based on the evaluation results four important reasons can be distinguished:

- compilation of data, information and knowledge from various areas (thematically and geographically) and quick access to it;
- participation, explanation and justification in the planning and decision-making process;
- support of planning and decision-making process by generating special insights, assessing impacts and measures and so forth; and

- handling of the complexity of IRBM and better understanding, including future perspectives.

Potential users were asked: 'What are key functionalities for a DSS from your perspectives?' The interlocutors considered: (i) building and exploring 'what-if' scenarios; (ii) assessing the diagnosis; and (iii) showing alternatives to meet management goals as the most important functionalities. This includes aspects such as analyzing and visualizing interrelations of land-water-ecosystems, cause-effect-interrelations, assessing scenarios and evaluation of management options (e.g. cost-effectiveness analysis of combined measures). Furthermore, goal seeking and visualisation of information were reported as important functionalities. The modelling aspect in DSS is generally greatly appreciated. However, live simulation is not necessarily required; using pre-calculated simulations is also an option. Another crucial factor for the success of a DSS is the way in which the DSS is implemented into the 'real' working environment of the user.

6.6 Techniques for Model Integration

The techniques used for technical model integration in the development of the *Elbe DSS* can be grouped into three categories – strong coupling, weak coupling and reimplementation.

Strong coupling includes techniques where an explicit software interface between model code and the simulation engine is created, which means that the simulation engine directly calls functions in the model code. Applying a strong coupling technique is possible with or without having access to the source code of the model. If the source code of the model is available and it is written in the same programming language as the ISDSS (C++ in case of the *Elbe DSS*), it can be integrated directly at the source code level. If the source code is available, but is written in a different programming language, this gap can often be bridged by applying special inter-language support libraries (e.g. Boost.Python, a C++ library which enables seamless interoperability between C++ and the Python programming language) or component middleware (e.g. COM, CORBA, .NET). In case the model code is only available as a binary component (e.g. dynamic link library or.NET component), strong coupling can still be applied. However, this requires the model code to feature the necessary software interfaces to connect it to the model controller of the ISDSS and possibly other model blocks in the ISM. Since this is unlikely for any model code that existed before the ISDSS started, having access to the source code gives the technical model integrator the most options by far.

The main advantage of the strong coupling approach is that, due to the lack of overhead and in memory access to data structures, it holds the potential to achieve high performance and good flexibility of the resulting ISM. The main disadvantage is that it creates complicated technical requirements for the model code, which may lead to costly modifications to the model code.

Weak coupling is the method of choice when strong coupling is not feasible due to technical, budgetary or organizational constraints. Instead of directly accessing in-memory data structures with weak coupling, the simulation engine uses the file system to communicate with a model. This approach is usually chosen when the model we want to integrate in the ISDSS is only available as an independent executable, which cannot be modified by the technical model integrator. In this case, the ISDSS first prepares an input file for the program that contains the model, and then issues an operating system call to start a separate process that runs the model program. After the model program has finished, the ISDSS reads the output file and applies the results to the internal data structures. The code in the ISM that controls the external model program takes the role of a proxy for the external model. It therefore features the necessary interfaces to connect to other model blocks in the ISM and the simulation engine.

The technical integration of the *GREAT-ER* and *HBV-D* models (Berlekamp *et al.* 2007) in the *Elbe DSS* are examples of the weak coupling approach with communication via the file system. Weak coupling techniques put very few requirements on the model we want to integrate. Therefore, their main advantage is that they allow the integration of models that otherwise could only be integrated by re-implementing them. The main drawback of weak coupling is that, due to slow file system operations, it often creates a performance bottleneck in the system. The dependence of the proxy on the external model program may also increase the complexity of maintenance of the system. A special form of weak coupling is 'asynchronous integration'. When a model is computationally too demanding to be integrated in an interactive application, one of the choices is to calculate some results offline and then integrate them as a scenario in the ISDSS. The climate scenarios in the *Elbe DSS* are an example of this. Asynchronous integration gives the ISDSS user access to the original model (e.g. via a web service) outside of the execution context of the ISM and thereby to add new scenarios.

Reimplementation of a model offers the most flexibility for the technical model integrator. In many cases, reimplementation yields the cleanest model code and the best possible performance. In the *Elbe DSS*, some of the flooding related models were delivered to the project as MATLAB® code (http://www.mathworks.com/products/matlab/) and then re-implemented in C++. In this particular case, the execution speed could be increased by a factor >100. For small models, reimplementation can be the most economical way to integrate the model in the ISDSS, because the code that would be necessary to 'wrap' the original model code can quickly become more complex than the model itself. The downside of the reimplementation approach is that it can make the maintenance of the code more complex. If the original author of the model decides to further develop the original model code instead of the reengineered code, two implementations of the same model have to be kept synchronized.

Table 6.3 shows for three models that were integrated in the *Elbe DSS* a comparison of the amount of code required for the model integration versus the model code itself. The last column in the table gives an estimation of the performance gain if the model code would be reengineered in C++.

Table 6.3 Proportion of model code versus code for model integration of selected models in the *Elbe DSS*

Model	Model code (# of lines)	Code for model integration (# of lines)	Potential performance gain with reengineering of model code in C++
MONERIS	(Excel model)	10,000	5x–10x
HBV-D	5,300	4,000	2x–5x
GREAT-ER	6,800	7,000	5x–20x
Total	–	21,000	–

6.7 Features of the *Elbe DSS*

6.7.1 Conceptual Modelling

A major goal of the feasibility assessment for the *Elbe DSS* was to clarify what is wanted and what is possible. Consultations with potential end users, together with a thorough analysis of results available from previous research projects on the Elbe River, yielded a description of the system scope and a preliminary set of functional requirements. On this basis, a *conceptual system model* (CSM) was developed in the first phase of the pilot project. The CSM identifies the processes, measures, indicators and external scenarios considered relevant for the policy context at hand, as well as the cause-effect relationships and information flows between these system elements. Depicted in a qualitative system diagram (Fig. 6.3; De Kok *et al.* 2008), the CSM proved to be a useful design artefact that: (i) effectively supported the communication between scientists; developers and stakeholders; (ii) formed the blueprint for the technical design of the system; and (iii) helped to monitor the progress of the conceptual system design during the development iterations of the project.

In the diagram shown in Fig. 6.3, system elements are connected by arrows which indicate cause-effect-relationships or information flows between the elements. The thick frames with the word ‘System’ represent the system boundary which separates system elements from their external inputs. External factors like external scenarios, measures and management objectives are linked to those system elements where they take effect. The CSM of the *Elbe DSS* is structured in four linked modules (catchment, river network, main channel, river section), each of them representing a specific spatial scale. Another useful technique to manage complexity is hierarchical composition, where a system element at one level may represent a full CSM at a lower level.

The qualitative system diagram is an effective communication tool for end-user and stakeholder participation. Early in the development process, particularly before any computer code is written, it shows what the ISDSS will represent and what it will not. In a collaborative effort between scientists and end users, the CSM is refined in several iterations, until mutual agreement is reached that the CSM: (i) meets the

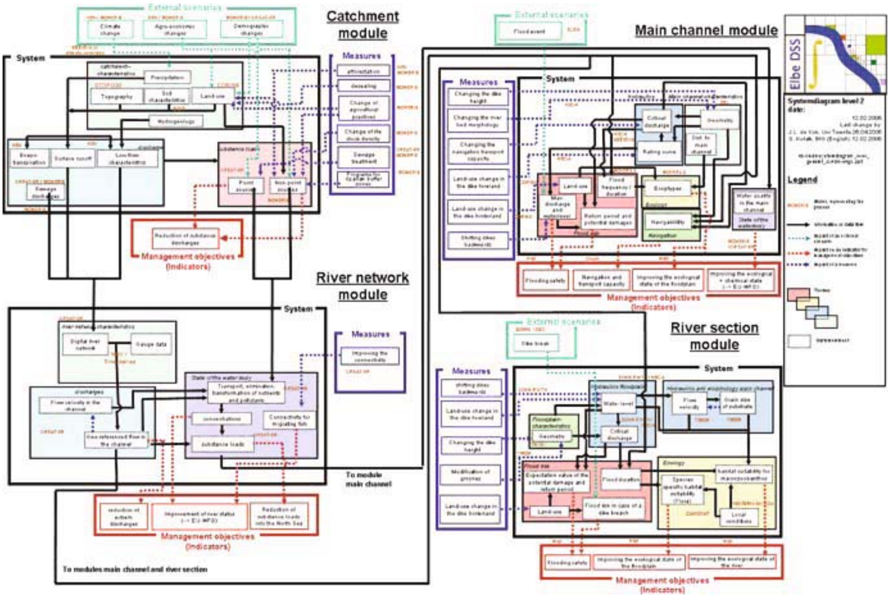

Fig. 6.3 Qualitative system diagram for the *Elbe DSS* (De Kok *et al.* 2008)

user requirements; and (ii) can be implemented with the available models and data. In the next step, models and data have to be associated with each system element. In the *Elbe DSS*, the qualitative system diagram has become part of the software as an interactive hierarchical diagram that acts as the central navigation instrument of the graphical user interface (Fig. 6.4). Scientists and modellers often prefer this system oriented type of interface; however, the results of the evaluation with end users (see below) indicated that they might be better served by a more sequential and task oriented approach to the user guidance. For example, the user interface of a river DSS may be based on a map showing the geography of the river basin.

6.7.2 System Architecture

In numerous projects, the authors have learnt to recognize generic components and architectural patterns that are common to many ISDSS. This has led RIKS to develop GEONAMICA®, an object-oriented application framework for ISDSS. While a decade ago the ISDSS developer had to start almost from scratch, nowadays mature application frameworks (e.g. GEONAMICA®, *SME* and others) can be utilized for time and cost efficient development of modular and flexible ISDSS applications. Fayad (1999) describes a framework as a reusable design of a system that describes how the system is decomposed into a set of interacting components. The

Fig. 6.4 Graphical user interface of the *Elbe DSS* (See also Plate 15 in the Colour Plate Section)

framework describes the component interfaces as well as the collaboration patterns between the components by means of special purpose textual (IDL, OCL) or graphical (UML) languages. Besides an abstract design specification, application frameworks may also provide a partial implementation in the form of a skeletal application that can be specialized to produce custom applications. Object-oriented frameworks are designed and implemented as a collection of collaborating abstract classes. They make use of the basic principles of object technology as described in Graham (1998): 'encapsulation' – data and processes are combined in objects and hidden behind an interface; 'polymorphism' – the ability to use the same expression to denote different operations; and 'inheritance' – implements the idea of classification and represents a special case of structural inter-relationship between a group of objects.

Figure 6.5 presents a simplified architectural overview of an ISDSS application based on the *Geonamica*® framework. *Geonamica*® has been used as the technological basis for the *Elbe DSS* and many other ISDSSs such as *RamCO, SImLucia* (Engelen *et al.* 1998a), *WadBOS* (Huizing 1998), *Environment Explorer* (Engelen *et al.* 1998b), *Modulus* (Engelen *et al.* 2000), *Murbandy* and *MedAction* (van Delden *et al.* 2007).

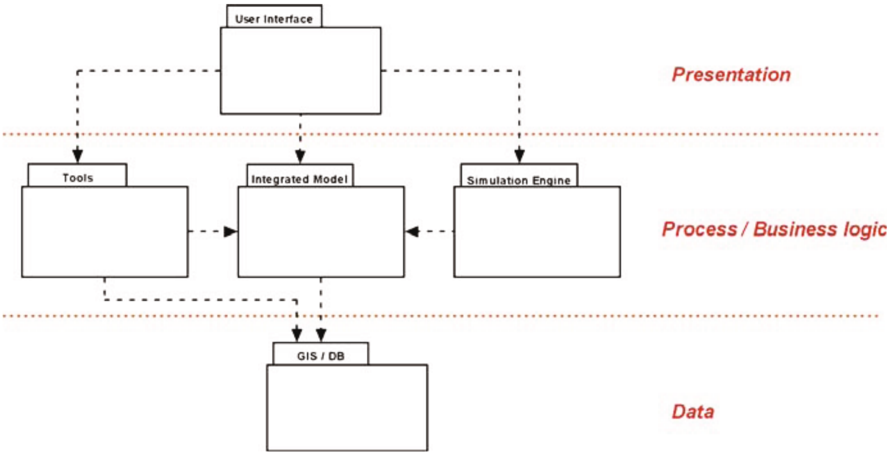

Fig. 6.5 Simplified reference architecture of a GEONAMICA® ISDSS application

6.7.3 *Iterative Development*

Requirements are seldom well understood at the beginning of an ISDSS project. Users usually start to recognize and articulate what they really want at the time when developers provide them with the first working prototypes of the product. From then on, requirements will never stop changing and evolving for the rest of the project lifetime and beyond. Acceptance of the fact that it is impossible to eliminate requirements' ambiguity and continuous change has lead to a shift from the classic linear lifecycle models (e.g. waterfall model) to more iterative and incremental approaches to ISDSS development. For the development of the *Elbe DSS*, we used the 'Evolutionary Delivery' lifecycle model as described in McConnell (1996). After an initial linear phase (top left-hand boxes in Fig. 6.6) that defines the core architecture of the system, it enters in an iterative and incremental cycle that elicits end-user feedback in joint workshops and delivers working prototypes at regular intervals.

This approach provided the development team with sufficient flexibility to adopt mid-course changes to the requirements (e.g. increased importance of flooding prevention functionality due to catastrophic events in 2002 or including policy options related to the recent climate change debate) and thereby increasing the relevance of the product for the potential end users.

6.8 Evaluation of the *Elbe DSS* Prototype

An important part of the ongoing requirements development process was the evaluation of the *Elbe DSS* prototype with end users. When the prototype became available by the end of 2005, an evaluation with users was carried out in close cooperation with BfG

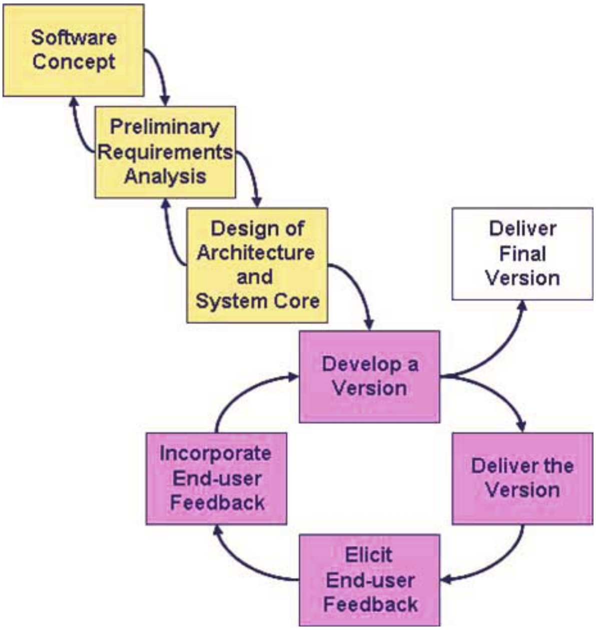

Fig. 6.6 'Evolutionary Delivery' lifecycle model (after McConnell 1996)

and the developer team. The *Elbe DSS* is one of the first systems realised which focuses on the complex field of IRBM for the strategic level of management. This evaluation served as a case study to analyse requirements for such different matters as system requirements, displayed system contents, data and models, decision and management support functionalities, performance, (sustainable) data management and maintenance. A detailed description of the *Elbe DSS* evaluation is given in Evers (forthcoming).

The evaluation was realised by means of user tests. As methodology, user tests with hands-on exercises, observations recorded in writing and questionnaires were undertaken. Furthermore, user test and system validation/verification were carried out both by a group of students and by one of the authors. During the winter term 2006–2007, an evaluation of the *Elbe DSS* prototype was undertaken with five different user groups who are concerned with IRBM on the strategic level. Test users were selected from the following organisations:

- a federal environmental institute;
- a sectoral management institute for coastal protection, water management and nature conservation in one of the Länder (federal states);
- a river-basin organisation of the River Elbe;
- an engineering company which is specialized in environmental planning; and
- a non-governmental organisation with focus on integrated water management.

The preliminary evaluation results can be summarised in the following points.
Key success/failure factors:

- Definition of purposes and requirement elicitation is a key issue. Thorough elicitation of requirements and users to define the purpose of the software is

essential rather than the software defining users' purpose. The system must have a positive effect on users' work (environment) and beyond, e.g. increase efficiency/quality.
- Definition of the diverse resources is crucial to assure continuous development, to ensure that enough data are available and to ensure maintenance of the system which is hardly ever finished.
- Active involvement of users, as already described above, is a key element of the development process.
- End user training sessions have to be provided.
- Prototyping of the system developed and tested by users involving realistic exercises. This gives tangible feedback before systems finalisation.
- The user-interface has to be oriented along users' technical backgrounds and should have integrated context-sensitive dialogues.
- The system has to be transparent (see above) with good documentation or library.
- The system should avoid fake accuracy and give information about the grade of correctness.
- It is important to consider the institutional and normative context of the institution where the software is supposed to be implemented. It has to be checked whether this institution has appropriate structure to use a PSS or can the system really depict the structure of that institution?
- Integration of evaluation during the development process is also an important point for successful implementation (see also next paragraph).

Knowledge integration: The elements of IRBM (despite groundwater management) are well integrated and the data pool is of good quality. The realisation meets the scientific requirements concerning models and methodologies.

User interface: A striking fact is that usage is complicated. The user gets lost easily and does not know what she or he can or has to do.

Decision-making process: The features of scenario simulation are highly appreciated. For example, the external scenarios with climate change and land-use change were considered to be very useful. Furthermore, the possibility of economic analyses is very welcome. Nevertheless, the usage of the scenarios was perceived as complicated. Shortcomings are evident in features like assessment, comparison of alternatives and optimisation of measures. These features are important for the decision-making process.

Transparency: This factor is regarded as very important. On the one hand, its realization can be regarded as very good as the test user assesses the scientific quality and data sources as good enough. On the other hand, the description of the content and the processes when operating the system was criticised. After the tests, all users were generally interested in the system.

Individual problems/aspects: The greatest interest in using the system was demonstrated by the members of the non-governmental organisations (NGOs), the federal environmental agency and the engineering company. The representatives of the institutions which are directly involved with operational management in IRBM, such as the regional water management authority and the river basin community, doubted whether they would actually use the *Elbe DSS* since their daily management demands are more of sectoral than integrative character.

Feedback by test persons concerning potential users of the Elbe DSS: was named with engineering companies (4), regional and federal authorities (3), a university, a river basin organisation, a water board, a sewage plant constructer and (diverse) institutions in nature conservation, agriculture, flood management and water management (each one nomination). This result uncovers a gap between the potential users as foreseen by the developers and the test users' assessment.

Against the background of the preliminary evaluation results, the question as to whether an ISDSS can provide useful support for IRBM can be answered affirmatively. The perceived added value can be summarized as the functionalities: developing a better insight of cause-effect-relations and getting a 'bigger picture' of relevant issues, relationships and optional measures.

For a generic *IRBM DSS* we can identify the following types of organisations as the most likely potential users:

- federal institutions which operate on a meta-management level;
- NGOs which focus on IRBM and are interested in gaining better insight into processes and optional management;
- engineering companies which want to generate specific information and data; and
- possibly universities to educate students in IRBM (this group did not take part in the DSS evaluation but one of the authors has practical experience in this field).

6.8.1 Future Trends

A standalone PC application is not the optimal system architecture for all ISDSS applications. The models and data in an ISDSS are often delivered by many different institutions. Knowledge management may become less complex, when these institutions remain in control and take the responsibility for the maintenance of their contributions to the ISDSS. Computationally demanding models are often better run on dedicated computers, which are optimally configured for this task. Keeping a tightly integrated PC application synchronized with quickly changing functional and organizational requirements for the ISDSS can be complex and costly. In view of participatory approaches to planning and policy design, the accessibility of the ISDSS for a wide class of users becomes a key issue. A service oriented architecture, where the ISDSS is an online application orchestrated from various web services offered by the knowledge and data providers, may in some cases be more flexible and lower the costs.

Environmental data that are used in an ISDSS often are monitored at regular intervals by ground-based observation networks or remote sensing instruments. In case of the *Elbe DSS*, integrating these data involved a lot of handwork by scientists and GIS specialists. For future ISDSS applications, especially when intended to support management oriented tasks, we envision enhanced integration with, and more direct access to environmental monitoring systems.

6.9 Conclusions and Future Directions

The experience with the design of the *Elbe DSS* leads us to the following recommendations. Firstly, both the design process itself and the practical application require a balance between users requirements, the quality of the scientific content, and sound software engineering. During the project the emphasis can temporarily pertain to a single aspect, but the final product should be based on a balance between all three aspects (De Kok *et al.* 2008). The key to achieve this goal is to start from real policy questions and to put together a multidisciplinary development team with an open communication culture, consisting of end-users, scientists and IT-experts as equal partners.

Secondly, effective communication between the developers and with stakeholders concerning the state of progress and possibilities of a tool is difficult when the tool is still being developed. A qualitative system diagram can be very useful in this respect. Furthermore the role of an ‘architect’, with knowledge integration and communication management as main responsibilities, turned out to be an important success factor for ISDSS development. The architect is a generalist and a true integrator, taking responsibility for linking the work of the modellers, assuring quality control of the mathematical core (model base) of the DSS and bridging the methodological and knowledge gaps between end users, modellers and IT specialists.

Thirdly, flexibility to cope with different measures, priorities or unexpected conditions is essential for relevance and acceptance of a tool. This calls for the application of relatively simple and flexible models. A modular system architecture, based on an object-oriented application framework helps to achieve a system that can be maintained, extended and adapted for future needs at relatively low costs.

The institutional embedding can be furthered by early anticipation in the implementation phase and providing key stakeholders with responsibilities for the design and financing, including maintenance.

Trans-national data harmonization and the development of metadata standards for the domain of strategic river basin planning are prerequisites for effective and cost efficient knowledge integration and ISDSS development. The fact that in the development of the *Elbe DSS*, nearly any model used its own *ad hoc* data format clearly showed how much needs to be done in this area. A lot of code needed to be developed just to translate between different representations of basically the same concept (e.g. a river network).

How to technically handle and, perhaps even more important, communicate uncertainty related issues in complex integrated models is still an open question. Solutions to this question could help to increase the acceptance of ISDSS as a tool for real world planning and policy making.

Finally, the interdisciplinary, integrated and holistic approach that underlies ISDSS often is ahead of the actual planning and policy making practice, where responsibilities and power are still organized in the classical sectors with only limited collaboration between sectors. Policy directives like the EC Water Framework Directive (EU 2000) and examples like the *Elbe DSS* may trigger further research into integrated methods for policy design, strategic planning and management of our natural resources.

References

Batty, M. (2007) Planning support systems: progress, predictions and speculations on the shape of things to come, Paper presented to the Seminar on *Planning Support for Urban and Regional Analysis*, Lincoln Institute of Land Policy, Cambridge, MA.

Berlekamp, J., Graf, N., Hess, O., Lautenbach, S., Reimer, S. and Matthies, M. (2007) Integration of MONERIS and GREAT-ER in the Decision Support System for the German Elbe River Basin, *Environmental Modelling and Software*, 22(2): 239–247.

Borshchev, A. and Filippov, A. (2004) From system dynamics and discrete event to practical agent based modelling: reasons, techniques, tools, in *Proceedings of the 22nd International Conference of the System Dynamics Society*, 25–29 July, Oxford.

Campagna, M., De Montis, A. and Deplano, G. (2006) PSS design: a general framework perspective, *International Journal of Environmental Technology and Management*, 6(1/2): 163–179.

De Kok, J.L., Kofalk, S., Berlekamp, J., Hahn, B. and Wind, H. (2008) From design to application of a decision-support system for integrated river-basin management, *Water Resources Management* (available online). http://dx.doi.org/10.1007/s11269-008-9352-7

Delden (van), H., Luja, P. and Engelen, G. (2007) Integration of multi-scale dynamic spatial models of socio-economic and physical processes for river basin management, *Environmental Modelling and Software*, 22(2): 223–238.

Denzer, R. (2005) Generic integration of environmental decision support systems – state-of-the-art, *Environmental Modelling & Software*, 20(10): 1217–1223.

Engelen, G., White, R. and Uljee, I. (1998a) Vulnerability Assessment of Low-Lying Coastal Areas and Small Islands to Climate Change and Sea Level Rise – Phase 2: Case Study St. Lucia, Report to the United Nations Environment Program, Caribbean Regional Coordinating Unit, Kingston, Jamaica, RIKS publication, Maastricht.

Engelen, G., White, R. and Uljee, I. (1998b) De Leefomgevingsverkenner. 'Proof of Concept' – Versie van een Integral Model voor het berekenen van het Leefomgevingskapitaal in Nederland, RIVM publication, Bilthoven.

Engelen, G., Meulen, Van Der M., Hahn, B., Mulligan, M., Reaney, S., Oxley, T., Mata-Porras, M., Blatsou, C., Kahrimanis, S., Giannoupoulos, P., Mazzoleni, S., Coppola, A., McIntosh, B. and Winder, N. (2000) *MODULUS: A Spatial Modelling Tool for Integrated Environmental Decision-making*, Final Report, Submitted to the European Union Directorate General XII Environment (IV) Framework, Brussels.

Evers, M. (forthcoming) Decision Support Systems for Integrated River Basin Management – Requirements for appropriate tools and structures for a comprehensive planning approach, Dissertation at the Faculty of Architecture and Landscape, Leibniz University of Hanover.

EU (2000) Directive 2000/60/EC of the European parliament and of the council of 23 October 2000. Establishing a framework for community action in the field of water policy, *Official Journal of the European Communities*, L327, 1–72. http://europa.eu.int/comm/environment/water/water-framework/index_en.html

Fayad, M.E., Schmidt, D.C. and Johnson, R.E. (1999) *Building Application Frameworks: Object-Oriented Foundations of Framework Design*, Wiley Computer Publishing, New York.

Graham, I. (1998) *Requirements Engineering and Rapid Development*, Addison-Wesley Longman, Harlow.

Hahn, B. and Engelen, G. (2000) Concepts of DSS systems, in German Federal Institute of Hydrology, *Decision Support Systems (DSS) for River Basin Management*, Koblenz, pp. 9–44.

Harris, B. (1989) Beyond geographic information systems: computers and the planning professional, *Journal of the American Planning Association*, 55(4): 85–90.

Huizing J.J., Engelen G., Ven van de K., Pothof, I. and Uljee I. (1998) *WadBOS: een prototype van een kennissysteem voor beleidsanalyse van de Waddenzee: eindrapport*, Rijkswaterstaat Directie Noord-Nederland, Harlem.

Kok, K. and van Delden, H. (2007) Combining two approaches of integrated scenario development to combat desertification in the Guadalentin watershed, Spain, *Environment and Planning B*, *34 (in press).*

Marakas, G.M. (1998) *Decision Support Systems*, Prentice Hall, Upper Saddle River, New Jersey.

McConnell, S. (1996) *Rapid Development*, Microsoft Press, Redmond.

Mintzberg, H., Raisinghani, D. and Théorêt, A. (1976) The structure of “unstructured” decision processes, *Administrative Science Quarterly*, 21: 246–275.

Walker, D.H. (2002) Decision support, learning and rural resource management, *Agricultural Systems*, 73: 113–127.

Westmacott, S. (2001) Developing decision support systems for integrated coastal management in the tropics: Is the ICM decision-making environment too complex for the development of a useable and useful DSS? *Journal of Environmental Management*, 62: 55–74.

Additional Reading

Costanza, R. and Voinov, A. (eds.) (2003) *Landscape Simulation Modelling: A Spatially Explicit, Dynamic Approach*, Springer-Verlag, New York.

Seppelt, R. (2003) *Computer-Based Environmental Management*, Wiley-Verlag GmbH & Company KgaA, Weinheim.