A methodology for the design and development of integrated models for policy support

H. van Delden\textsuperscript{a,}\textsuperscript{*}, R. Seppelt\textsuperscript{b}, R. White\textsuperscript{a,c}, A.J. Jakeman\textsuperscript{d}

\textsuperscript{a}Research Institute for Knowledge Systems, P.O. Box 463, 6200 AL Maastricht, The Netherlands
\textsuperscript{b}UFZ – Helmholtz Centre for Environmental Research, Department of Computational Landscape Ecology, Leipzig, Germany
\textsuperscript{c}Memorial University Newfoundland, Newfoundland, Canada
\textsuperscript{d}Fenner School of Environment and Society and National Centre for Groundwater Research and Training, Australian National University, Canberra ACT 0200, Australia

**Abstract**

The development of Decision Support Systems (DSS) to inform policy making has been increasing rapidly. This paper aims to provide insight into the design and development process of policy support systems that incorporate integrated models. It will provide a methodology for the development of such systems that attempts to synthesize knowledge and experience gained over the past 15–20 years from developing a suite of these DSSs for a number of users in different geographical contexts worldwide.

The methodology focuses on the overall iterative development process that includes policy makers, scientists and IT-specialists. The paper highlights important tasks in model integration and system development and illustrates these with some practical examples from DSS that have dynamic, spatial and integrative attributes.

Crucial integrative features of modelling systems that aim to provide support to policy processes, and to which we refer as integrated Decision Support Systems, are:

- Synthesis of relevant drivers, processes and characteristics of the real world system at relevant spatial and temporal scales.
- An integrated approach linking economic, environmental and social domains.
- Connection to the policy context, interest groups and end-users.
- Engagement with the policy process.
- Ability to provide added value to the current decision-making practice.

With this paper we aim to provide a methodology for the design and development of these integrated Decision Support Systems that includes the 'hard' elements of model integration and software development as well as the 'softer' elements related to the user-developer interaction and social learning of all groups involved in the process.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Integrated Decision Support Systems (DSS) are rapidly gaining attraction in the planning and policy-making community. When introduced into the decision-making process in a controlled way, they can create high added value by bringing scientific knowledge to the decision makers' table. Despite this high interest, only a few DSS are in actual use to support policy development and analysis. Academic literature recognizes several reasons for this, most notably a lack of transparency, inflexibility and a focus on technical capabilities rather than on real planning problems (Uran and Janssen, 2003; Vonk et al., 2005; Geertman, 2006). In order to deploy a DSS as an instrument for strategic policy making, it has proven to be crucial that the system matches the perceptions, experiences and operational procedures of the policy makers and that it enhances their current policy practices rather than replace existing and well-embedded ones (Van Delden, 2003; Van Delden et al., 2007; McIntosh et al., 2007; Te Brömmelstroet and Bertolini, 2008).

For integrated models to be able to provide useful support to policy making, they should be able to represent the complex
interactions taking place in the human—environment system. Over the past decade the science on model integration has gained in importance and has been facilitated by the improved software capabilities that allow for the development of DSS featuring integrated models. Integrated modelling has evolved from the early integrated assessment models (Rotmans et al., 1990) and ecological economic models (Low et al., 1999) to the current spatially explicit and complex systems as described, for example, by Forsman et al. (2003) on linking farmer’s behaviour with air and water quality, by Aman et al. (2004) on air pollution and by Van Delden et al. (2007) on regional development and desertification. This paper builds on practical experience of Integrated Spatial Decision Support Systems (ISDSS) development over the past decades but much of the discussion is also relevant to systems with modest or no spatial representation. It provides a methodology for the design and development of integrated DSS that includes both ‘hard’ and ‘soft’ factors. Hard factors relate to the selection and development of a model, model integration, model evaluation and the selection of the software platform. The ‘softer’ factors relate to linking scientific knowledge to information relevant to policy support, emphasis on social learning of the different groups involved, the role of champions and the implementation of DSS in (policy) organisations. The need for this kind of approach in which both factors are incorporated is also recognised by McIntosh et al. (2007), Van Delden (2009) and Volk et al. (in press). The methodology builds on principles of software engineering, product design and DSS development and incorporates techniques such as evolutionary design and rapid prototyping (Cross, 1994; Marakas, 1999; Robertson and Robertson, 1999). It incorporates elements from the domain of integrated assessment modelling (IAM) by including multiple issues and stakeholders, integrating the human and the natural sciences and by incorporating multiple scales of system representation, spatial and temporal behaviour and cascading effects (Rotmans and Van Asselt, 1996; Parker et al., 2002; Jakeman and Letcher, 2003). The methodology makes use of interaction design methods (Gullikson et al., 2003; Moggridge, 2007) to ensure a user-centred and demand-driven approach and builds on psychology and organizational theory (Weick, 1979; Langley et al., 1995) to understand the process of decision-making, providing insight into both the co-creation of DSS by developers and users and its implementation in organisations. Although literature on the design and development of decision support and other software systems is already widely available we aim to provide added value to this by focusing explicitly on the core characteristics of developing integrated models for policy support: the science-policy interface, the integration of models from different disciplines and the collaborative effort of users, scientists and software developers.

2. Integrated models for policy support

Integrated modelling systems for policy support can be found under a diversity of names, amongst others (Spatial) Decision Support Systems or (S)DSS (Turban, 1995), Planning Support Systems (Geertman and Stillwell, 2003) and Policy Support Systems or PSS (Van Delden et al., 2007). Although they differ in their specifics, for the purpose of this paper we will group them all under the name of Decision Support Systems (DSS) since they have sufficient characteristics in common (for examples of good recent discussions see, e.g. Pettit, 2005; Pettit and Klosterman, 2005). Several authors have mentioned the following common characteristics:

- able to support policy-relevant questions (Parker et al., 2002; Geertman and Stillwell, 2003; Van Delden et al., 2007),
- pay particular attention to long-term problems and strategic issues (Geertman and Stillwell, 2003; Van Delden et al., 2007),
- aim to explicitly facilitate group interaction and discussion (Geertman and Stillwell, 2003; Newham et al., 2006),
- apply in complex and ill-structured or wicked decision domains, characterised through a large number of actors, factors and relations, a high level of uncertainty, and conflicting interests of the actors involved (Rittel and Webber, 1973; Van Delden, 2000; McIntosh et al., 2007),
- are user friendly in entering input, viewing output and analysing results (see for instance Volk et al., 2007, 2008),
- incorporate actual data and process knowledge from different disciplines (Van Delden et al., 2007),
- operate on different scales and resolutions where required (Van Delden et al., 2007; Volk et al., in press),
- may be fully dynamic with feedback loops between individual models (Van Delden et al., 2007, 2008a),
- built as a flexible component-based system that can be extended with additional modules over time (Argent, 2004; Van Delden et al., 2009).

The methodology proposed in this paper is developed for the design and development process of DSS with the above-mentioned characteristics. Although the DSS described in this paper mostly focus on dynamic spatial simulation models that integrate biophysical and socioeconomic model components, the methodology is not limited to this type of systems. It can also be applied to systems that include for example non-spatial models, such as Bayesian networks (Farmani et al., 2009; Ticehurst et al., 2007), models that calculate an end-condition rather than simulating temporal dynamics, or models that focus on optimisation instead of simulation (Seppelt, 1999; Seppelt and Voinov, 2002, 2003).

The examples in this paper come from DSS that were developed using the Geonamica software platform for spatial modelling and (geo)simulation (Harkens et al., 2008). Geonamica has been the basis for many DSS that vary greatly in their application domain (urban and rural areas, coastal zones, river basins) and spatial extent (cities, countries, EU-27), based on the requirements of the user. Examples of DSS developed with Geonamica are WadBos (Engelen et al., 2003a), Environment Explorer (Engelen et al., 2003b), MedAction (Van Delden et al., 2007) and its predecessor MODULUS (Oxley et al., 2004), DeSurvey Integrated Assessment Model (IAM) (Van Delden et al., 2009), Xplorah (Van Delden et al., 2008a), Elbe-DSS (De Kok et al., in press), LUMOCAP (Van Delden et al., in press), WISE (Rutledge et al., 2008) and MOLAND (Barredo et al., 2003). Although different in their application domain, they are built on very similar principles. They feature complexly linked multi-scale, spatial-dynamic models and have a user interface that enables interactive access to all drivers and individual models representing the processes. Thus, the user may enter and change driver or parameter values to specify their inputs, can invoke tools for the analysis and visualisation of the model outputs, and can access the integrated help system which clarifies the underlying assumptions and formal definitions of the models and the data used.

In Section 4 of the paper, examples will be provided from the development process of the following DSS:

- The MedAction PSS and its successor the DeSurvey IAM, aiming to provide support to regional development and desertification and having a main focus on sustainable farming, water resources and land degradation in arid and semi-arid regions (Van Delden et al., 2007, 2009; Kok and Van Delden, 2009; RIKS, 2009a).
The WISE ISDSS, aiming to support the development of a strategic vision for the Waikato region in New Zealand by taking into account social, environmental and economic well-being (Rutledge et al., 2008; Huser et al., 2009; Creating Futures, 2009).

The LUMOCAP PSS, aiming to assess the impact of the Common Agricultural Policy (Commission of the European Communities, 2009) on the land use and landscape of the 27 countries of the European Union (Van Delden et al., in press; JRC, 2009).

The Xplorah SDSS, aiming to support spatial planning on the island of Puerto Rico by focusing on economy, population, land use and transport in an integrated manner (Van Delden et al., 2008a; RIKS, 2009b).

The Elbe-DSS, aiming to support the implementation of the European Water Framework Directive in the Elbe catchment in Germany (De Kok et al., 2009; Lautenbach et al., 2009; Hahn et al., 2009).

The examples provided above vary in their level of complexity (number of models included, number of model interactions and detail of processes modelled), spatial and temporal resolution and spatial extent. All systems except for the Elbe-DSS are dynamic spatial simulation models. The Elbe-DSS calculates one final state. Whenever reference is made in this paper to specific documents other than the journal articles, they can be downloaded from the project or product websites (RIKS, 2009a; Creating Futures, 2009; JRC, 2009; RIKS, 2009b).

3. Methodology for design, development and implementation

There are some important lessons learnt from the past development of several DSS. An ideal development process can best be described as an iterative process of communication and social learning amongst three involved parties (see, e.g. Engelen, 2000; Te Brömmelstroet and Bertolini, 2008) (Fig. 1).

First, there are the end-users of the system. In this paper they are often called policy makers, but these include all kinds of actors that are present in the process that is supported. Together, they provide the policy context and define the policy problems. They also have certain demands for the functions and usage of a DSS, which has to link to their personal (daily) working habits, educational background and professional paradigms. The iterative process enables these demands to be included in the DSS.

Second, there are scientists (or other experts) from a range of disciplines and responsible for correctly representing the main processes in their models, including the underlying assumptions and choices of scale, resolution and level of detail (the substance of the model). These choices are based on research and analysis that meets the rigorous standards of science, taking into account the problem definition, data, knowledge and resource constraints.

Third, there are IT-specialists who design the system architecture and carry out the software implementation of the models and user interface. This architecture has to enable the link between rigorous scientific substance and the demands of the policy makers and other actors.

For the development of complex DSS especially, a fourth role is vital for success. The DSS architect(s) has the main responsibility for integration, communication and management of the three other groups and assures the quality of the integrated model underlying the system. As a generalist, the architect bridges methodological and knowledge gaps between policy makers, scientists and IT-specialists — both between and within groups. In order to fulfill this role, the architect needs a solid and intuitive understanding of the application domain and the purpose of the system, as well as very good communication skills.

The interaction between the three groups involved is just as important, for the quality and usefulness of the final product, as the tasks carried out by each group individually. Policy makers and scientists select policy-relevant research and models capable of answering the problems set out by the policy makers, translate policy options and external factors into model input and translate model output into policy-relevant indicators. Scientists and IT-specialists work together to implement new models, link existing models and ensure consistency throughout the system. IT-specialists work with policy makers to set up a user interface that represents the relevant input and output in a comprehensible manner without overwhelming the user with the wealth of available information and possibilities.

The interaction helps each group to gain a better understanding of the needs of the others and the expertise that they offer. For this understanding to take full effect, an iterative approach is best suited (Fig. 2). Although we recognize that these steps are not as clear-cut and sequential as might be suggested in the figure and that several actions might take place simultaneously, for the sake of clarity we present them sequentially in the next section.

When the initial goals and requirements of the system have been established, IT-specialists set up a prototype system to help assess its usability and demonstrate to policy makers the possibilities. Their feedback is important to make the model more robust, add missing input or output to the system and improve the operability of the user interface. The prototype can be used by scientists to train policy makers — and their technicians — in the use of the system. This includes translating real-world problems or questions into interventions in the system and translating an analysis of model output into concrete, valuable recommendations or conclusions.

Besides giving scientists a better understanding of the policy-making process, such consultation can help increase the practical value of the system — that is, make sure that it will be used and that it will be used appropriately (see Section 4.4.2 for more details). Champions, power users who actively stimulate the uptake of the system in their organisation, can play an important role in this process. Due to its integrated approach, the utilisation of the DSS is likely to have an impact on the work-practice of the involved organisation. After some time, an extension of the system or

![Fig. 1. Main parties, responsibilities and integration issues during the development of a DSS.](image-url)
refinement of some part may prove desirable and the development process can enter another iteration.

4. Important tasks within the process

Within the general framework presented in Section 3 important tasks can be distinguished that have to be fulfilled during the design and development of a DSS. As stated before, these tasks need not — and often cannot — be carried out in a purely sequential order. Due to the complexity of the process and the innovativeness of it for most contributors, it is very unrealistic to have this expectation and therefore one should be able to go back to previous tasks, either to complement or detail them, or sometimes to change them completely. The tasks are presented in Fig. 3 and described in detail in the sub-sections below. Task numbers are provided in Fig. 3 and sub-sections in which the tasks are described follow these numbers.

4.1. Defining the scope

Early in the design and development process a focus needs to be found for the overall system. The focus will be a compromise between addressing today’s problems and making the system flexible enough to handle envisaged future needs. This focus is related to the context of the DSS as well as its intended use and is ideally guided by the users. It is the main guideline for the design and development of the DSS. Over time this focus can be further fine-tuned or adapted, but very often drastic changes in the scope result in major adaptations of all tasks that need to be carried out. On the other hand, experience teaches that it is hardly ever possible to define a full set of system specifications at the beginning of the process. An iterative process is required that allows for fine-tuning at later stages together with adaptations to the initial ideas once prototypes are developed and a better understanding of the possibilities and limitations is obtained. Scoping documents like the LUMOCAP strategy paper (JRC, 2009) and the WISE specifications document (Creating Futures, 2009) help in specifying the issues raised and decisions made and in communicating them to all involved parties. When updated throughout the design and development process, these documents can become an important backbone of this process. At the end of the scoping task it should be clear what added value the DSS could and should bring to the current practice and, given the time and financial resources for the development process, what is the best way to reach this goal.

4.1.1. Defining functions and use

In an ideal situation a user would come to the architect or development team with a request to develop a DSS and would already have the scope of the system in mind. However, in many cases DSS are developed from research funding opportunities and a potential user group is defined that has an interest in the system, but not yet a full commitment, desire or need to actually use the DSS when it becomes available. Even when users initiate DSS development they often do not have a clear vision for the intended use of the system. A complicating factor is that the potential group of users can be rather large and consequently include many different opinions on the best way forward. Therefore the main questions to be answered at the outset of the development process are: Who is going to use it? What for? And, what is the added value that the DSS brings?
In our approach we make the distinction between end-users and users. Where-end-users use the results of the DSS, users actually push the buttons and know how to work with the system. Users and end-users can be the same person, but it is also possible to have them within one organisation or spread over different organisations. An example of the latter is the use of the LUMOCAP system by the Joint Research Centre of the European Commission (a research organisation) and its end-use by the Directorate General for Agriculture and Rural Development of the European Commission (a policy organisation). Defining these organisations and, even more importantly, individuals in these organisations, helps to target the development, includes them in the social learning process and creates commitment for future use. Champions can greatly facilitate this process because of their knowledge of and contacts within the user organisations. If no champions are identified so far, it deserves attention to find them as part of this initial task.

Next, main functions of the DSS need to be decided upon as well as to where and how in the policy process the system can provide support (problem recognition, identification of alternatives, assessment of the impact of alternatives, consultation, communication, deliberation and/or implementation). Amongst others, possible functions include knowledge management, what-if analysis, structuring the policy process, finding optimal solutions given a set of constraints, communication to people involved in the decision-making process and communication to the broader public. Many DSS fulfil several functions, but these need to be made explicit so that potential conflicts can be discussed and overcome. When a DSS aims to be used for different functions, it is very possible that the requirements for these functions are not always the same and trade-offs need to be made. For example a DSS that will be used to communicate policy alternatives to a broad group of stakeholders will need a user interface that is easy to understand and focuses on policy-relevant elements, such as indicators, while for a DSS with detailed analysis capabilities a simple user interface does not have to be such an important requirement; a user interface that gives access to all data and parameters might be much more appreciated. Discussing these potential conflicts early in the development process and coming to an agreement about requirements is important to avoid later disappointment. The DSS described in Section 2 are mainly intended to support what-if analysis and to facilitate the communication of those involved in the decision-making process. Sometimes these were policy analysts and technicians from different disciplines within one organisation (e.g. the provincial government). Other times these were a wider group of stakeholders.

Finally decisions need to be made on the intended use of the system. This is partly related to the function, but requires more detail on the actual use of the system. One of the main questions here is whether the system is going to be used in workshops sessions or by individuals behind their desk? If the aim is to use it in workshops, then should the system be interactive and allow participants to try out different alternatives on the fly, or can pre-calculated results be used? Use in participatory workshop sessions allows an active involvement of end-users in developing, analysing and discussing different scenario alternatives. On the other hand, the running speed of the system can be limited which often is a compromise for the level of detail that can be included.

In our examples, WISE and Xplorah are well suited for use in participatory workshop sessions as they have a running time in the order of minutes for calculating developments 30–50 years into the future. The LUMOCAP system on the other hand has a running speed of several hours caused by its level of (spatial) detail and large area extent. Its results can still be discussed in workshop sessions, but an interactive use in such sessions will not be possible.

4.1.2. Defining policy themes, issues, options and indicators

DSS development is a very time-consuming and expensive task. Developing a system that can be used for a number of policy issues over a period of several years is therefore highly preferred over a system that can only be used once for one specific problem. Being too narrowly focused and making decisions on the design and development purely based on one question or problem, that is one application, can easily lead to a system that can only be used once. When including the broader context in which these types of problems occur and representing the system in which the problem ‘operates’, the reusability of the DSS becomes much larger. This also brings a developer to questions such as: do you implement a specific policy (such as the European Common Agricultural Policy (CAP)) or do you implement policy mechanisms like intervention prices, decoupled subsidies, etc. and thus include a higher level of abstraction with more flexibility? The mechanisms then relate to the CAP and allow you to model the CAP. In summary, being able to represent part of the system will enhance the ability to deal with a wider range of options, and hence provide more flexibility than a focus on the impact of one policy option on one policy indicator. One way of gaining such efficiencies is to focus on policy themes rather than individual problems. Themes are broader problem concepts that will remain important for policy analysis over the coming years. Within these themes fit several policy issues or problems of which some are pertinent at the moment, but others might become an issue in due time. An example is the MedAction PSS aimed at supporting policy impact assessment studies in the field of regional development and desertification. Broad themes defined in this context were sustainable farming, water resources and land degradation. The current issues related to these themes include the availability of water, how to price the water, and how to preserve the forests.

The next step is to find drivers that have an impact on the defined themes and issues. Drivers can be split into external factors, which the policy maker(s) cannot influence (e.g. climate, global markets and policies, technological developments) and policy measures they can implement or influence (e.g. market instruments like subsidies and trading, education stimuli, construction of infrastructure, zoning and other land use and water regulations). These drivers need to be explicitly or implicitly represented in the modelling system.

The final step is to define indicators that follow the main developments over time and provide some quantitative and/or qualitative measure of change in biophysical and socioeconomic outcomes relative to benchmark situations – the recent past being an oft-used benchmark. Biophysical outcomes might relate to potentially substantial impacts on the state of the environment, and socioeconomic ones to income levels, employment and cultural criteria, for example.

In considering this overview it is crucial to ensure a flow between the themes, issues, drivers and indicators and to discuss this flow and its individual elements thoroughly with the (potential) users. Often users will already have a good perception of the current problems, available measures and indicators and where gaps remain these can be filled through other sources, including the scientists involved and interest groups likely to be impacted. In situations where DSS are developed to support the analysis or implementation of a specific directive, this directive often provides valuable information for inclusion in the process. The Elbe-DSS for example was for a large part initiated to support the Water Framework Directive of the European Commission (European Commission, COM, 2000), and the DeSurvey IAM aims to support the implementation of the UNCCD’s National Action Programmes to combat desertification (UNCCD, 2009). Both documents provided ample information about the important indicators, which were
subsequently included in the systems. Van Delden et al. (2007) have provided the above-mentioned information in tabular form for the MedAction PSS.

Besides the creation of such a table, it is also important to decide on the scale and resolution that is desired by the user and supported by the science. The request for information at a certain level of detail has large implications for the selection of models as described in the next section. During the scoping task, meetings are recommended with the three groups involved, to discuss the feasibility of the user’s ideas and to make suggestions from scientific and/or software technical points of view.

4.2. Appropriate model selection

While users take the lead in the scoping task described under Section 4.1, scientists are generally in charge of this model selection task. The main aim is to model the processes at the level of detail that is required for policy analysis at the scale defined in the previous task. The starting point is the selection of the main processes and variables relevant for the themes, issues, drivers and indicators as addressed in the previous section. This task starts by setting up a conceptual diagram in which all main components and their linkages are included (see e.g. Van Delden et al., 2006a and examples of the conceptual diagrams of Xplorah and Med-Action, respectively). In constructing this diagram it is important to clarify the inputs and outputs of each model component and it might help to build cross-tables that show which flows are envisioned between the different components (see e.g. Van Delden et al., 2007). On the basis of the conceptual diagram individual models are selected to represent the relevant processes. These can either be existing models, or newly developed models. Furthermore links need to be created from the main drivers to the (scientific) models and from the models to the indicators. If it is not possible to provide the information required or link to the drivers agreed upon, the process needs to go back to the scoping task to decide if it is still relevant to pursue the development and if so what changes need to be made to the scope of the DSS.

In this task decisions need to be made on the complexity of the models, the number of variables, relations and processes modelled, as well as their appropriate temporal and spatial scales — extent and resolution. These decisions depend on a number of factors, of which the most important are:

- Intended use of the DSS and the requirements that follow from this. Choices will have to be made between simple process representations that allow the system to run fast and therefore promote its use in workshop sessions, or more detailed and accurate representations that require longer running speeds.
- Choice of scale, resolution and level of complexity required from a scientific point of view to be able to provide information on the scale and level of detail defined in the previous task. As indicated by Gibson et al. (2000) ‘Issues related to the scale of ecological phenomena are of fundamental importance to their study’ and hence for impact assessment models for policy support. The guiding principle regarding model complexity is not to be more complicated than necessary; that is not to introduce more (spatial or temporal) detail or (process and interactions) complexity than warranted for the purpose, while on the other hand not omitting crucial drivers and processes (Occam’s razor). Scaling issues have been discussed in various papers, amongst which are Ménard and Marceau (2005) regarding land use models, Booij (2003) regarding hydrological models and Mulligan (2004) regarding hydrology and plant growth models. When making decisions on the choice of scale, resolution and complexity, it is crucial to take the entire integrated model into account. It is important to ensure that the integrated system includes a balanced selection of models, because the system is only as strong as the weakest link. This may require simplifying some model components, which would not have been simplified in stand-alone applications. De Kok and Wind (2003) provide a further discussion about appropriate modelling.
- Availability of existing models that fit the purpose or can be adapted to fit the purpose, versus the possibility and need to develop new components. If models are available that fit the purpose or can easily be adapted to fit the purpose, this is preferred from the point of view of reusability. However, since individual models are often developed for a different purpose this might be a comprehensive task, making it easier sometimes to develop new components (see also Oxley et al., 2004). Also, when no models are available to simulate crucial processes, (simple) components will have to be developed. Several authors have published approaches to good modelling practices, amongst which are STOWA/RIZA (1999), Jakeman et al. (2006) and the followup demonstration papers by Robson et al. (2008) and Welsh (2008).
- Data availability. Much of the data required for the development of integrated systems is scarce. In general the request for a DSS implies that knowledge derived from certain available indicator systems need to be improved or gaps need to be filled. On the other hand, it may not be possible to replace missing data with a model. If limited data is available, selection of simpler process representations is preferred to avoid problems in setting up the model, calibration and validation (Mulligan, 2004; Van Delden et al., 2009). Quality and detail of the data available have direct impact on the quality and accuracy of the results. With limited data or data of poor quality, results will have a higher uncertainty and this should be taken into account in their interpretation. Depending on (the scale of) the policy question, the level of detail in the data will be more or less important.
- Time available to develop, set-up, calibrate and validate the application. Similar to choices made based on data requirements, it is important to take into account the available budget to set-up, calibrate and validate the DSS. If time is limited, it is best to develop or select relatively simple components that can be calibrated and validated within the available budget.
- Human capacity available to use the system. When the system is not intended to be used by experts, this sets limitations to the type of models and level of complexity that can be included. It is crucial that users are able to interpret results correctly. To be able to do this a good understanding of the models included is essential. The time and resources available for training play an important role in this decision.

An output of this task should be a document that describes all models included, in detail and in the same format. Using the same format helps to obtain a certain level of consistency and facilitates the model integration and implementation. Documenting the models in detail makes the way that the processes are modelled more explicit. The description of each individual component includes the aim of the model, its application domain, the assumptions and constraints, a description of the processes modelled, the equations or other representations used (e.g. rules) and a definition of the input, internal and output variables and parameters. This document often forms the basis of the final model descriptions. Examples of this format are the MedAction documentation and the WISE specifications document, available through the project web sites (RIKS, 2009a; Creating Futures, 2009).
4.3. Model integration

In the type of DSS described in this paper, the inclusion of dynamic feedback loops between model components is crucial in order to capture how systems might adapt when subject to change. This is certainly the case for the DSS presented in Section 2. An example is the dynamic interaction between land use and land suitability simulated in the MedAction PSS: when farmers irrigate with water with a high salinity, this has an impact on the suitability of the land for agricultural practices, which in turn impacts on the crop choices farmers make and the management techniques they will apply. This can ultimately lead to farmers abandoning their land, causing the plot to return to natural vegetation. Over time the salt may wash out, making the land suitable again for agriculture. In other situations it might be possible to handle such changes less automatically. For example, when states or outputs of a system change, it can be possible to modify the system in a subsequent run such as through a change to the scenarios or model parameters. This approach has the benefit of system changes being more explicit to the user.

The representation of feedback helps to pinpoint reinforcing effects and understand the (unwanted) side effects of proposed alternatives. In WISE, the ecological economic model provides the regional demands for land resources, while the land use model simulates the competition for space at the local level and feeds the ecological economic model with the land availability for different sectors based on this competition and the policy regulations (mostly zoning). When the supply of resources is lower than the demand, the economic model calculates the economic developments based on these resource limitations. Using an Input–Output approach, the impact of resource limitations becomes not only apparent in the sector for which the supply was limited, but also in all other sectors of the economy. Hence, protecting land from further development of particular economic activities might have unexpected reinforcing effects on the overall economy even in those sectors not directly related to the ones for which the policy was intended.

The task of model integration comprises both the coupling of the individual models and their software implementation. In doing so, we face some scientific and technical challenges. The main scientific challenges are related to:

- Dealing with models representing processes operating on different scales and having different spatial and temporal resolutions. In our approach a component model operates at the temporal and spatial resolution that best represents the process and ways are found to deal with these differences. An example of this is the MedAction PSS in which bio-physical models operating daily are integrated with land use and crop choice models operating yearly by using a yearly dynamic suitability component based on results from the bio-physical models. Aggregation of the daily results for soil moisture and temperature takes place according to an arithmetic mean over those periods important for crop growth, while for soil depth and salinisation values at the end of the year are used (Van Delden et al., 2007). More examples related to the scaling issues encountered in the DSS described in Section 2 can be found in Van Delden et al. (submitted for publication).
- Dealing with different types of models developed using different modelling paradigms (c.f. Seppelt, 2003). Many disciplines have their own specific way to construct a model and linking them is not evident. In developing coupled human-natural systems we often experience discrepancies between: (economic) models that are developed under an equilibrium assumption and calculate one ‘final’ state based at an equilibrium condition defined at a point in time; and (biophysical) simulation models that start with a set of initial conditions and simulate future development in subsequent time steps based on (a number of) drivers, never assuming an equilibrium, but always taking into account the forces of action and reaction and incorporating the accompanying feedback loops between them. Another example is the integration of transition models (such as land use change models) and models calculating a continuous development (such as growth of biomass or yield), which can cause (unwanted) shock effects in the integrated model. Whenever these shocks represent events that occur in reality (such as the conversion of an agricultural area into a residential area or a flood or fire event that takes place at a certain point in time), the integrated models are normally able to deal with these transitions. However, when transitions are mostly artefacts of the modelling approach (e.g. in models simulating transitions in natural vegetation type groups) such behaviour is undesirable (Van Delden et al., 2007).

Xplorah and MedAction are examples of DSS comprising integrated models including components operating on different spatial and temporal resolutions and based on different modelling paradigms. Specific discussion on temporal integration in the Elbe-DSS is provided by Lautenbach et al. (2009). DSS based on Artificial Intelligence approaches are quite mature in the way they integrate models and information based on different paradigms (e.g. Devesa et al., 2009; Comas et al., 2008).

Technical challenges are related to the development of a software platform that is able to integrate the individual components through dynamic feedback loops, is flexible to incorporate or eliminate models over time or for different simulation runs, allows interaction with the user and is able to provide fast running speeds. Since development of models is a very time consuming and expensive task, a modular approach that allows reusability of components is also preferred. In our examples we have made use of the Geonamica software platform (Hurkens et al., 2008). Selection of the appropriate platform is based on choices made in the previous tasks. Software platforms do not include any model components.

To facilitate reusability of components, DSS generators or modelling frameworks are available consisting of a software platform and a number of (model) components that can easily be configured to create specific DSS for particular problems. Argent et al. (2009) present a DSS generator for water quality modelling within which system users are able to select and link models, data, analysis tools and reporting tools to create specific DSS. A similar approach for land use modelling is the Metronamica framework (RIKS, 2009c) and for regional development the DeSurvey IAM modelling framework described in Van Delden et al. (2009).

A point often raised is that integrated models should not be too complex, especially when they have the aim to be used in a policy context. For integrated models similar principles are true as for individual models: reduction of complexity without omitting crucial components is in many cases the best solution. Nonetheless the final system might still be rather complex, since the real world system is inherently complex. Our experience has shown that policy makers are very capable of dealing with complexity, which is likely a result from their daily practice of operating in an interdisciplinary field with many different actors and processes. This might also lead to their preference for systems that include a rather high level of complexity (see questionnaire results from WISE and LUMOCAP on their respective websites), their main concern being the transparency of underlying assumptions, the way processes are modelled and making the uncertainty in the results explicit.
Over time, integrated models have a tendency to become larger and larger, driven by the desire of both modellers and users to specify processes and interactions in more detail and thus developing a more ‘realistic’ representation. Increasing the level of complexity, however, by including model components and/or modelling processes in a more detailed way also introduces higher maintenance costs for updating the data, calibration and validation, and does not always improve the understanding of the model behaviour and therefore can create problems in interpreting the results. Sensitivity analysis can help to eliminate non-crucial model elements. When the problem is not so much related to the introduction of unnecessary complexity, but rather that the running speed is no longer appropriate for interactive workshop sessions, simpler components or metamodels can be developed with a different level of complexity. In this case the simple version can be used for quick exploration during workshop sessions and the complex version for detailed analysis. Of course this imposes requirements on the flexibility of the software environment. It also requires a thorough explanation of the approach to the workshop participants, as results when running the system with the more complex components might differ from those obtained during the workshop. Xplorah includes this kind of flexibility and allows users to run selected municipalities at coarse (240 m) or detailed (30 m) resolution. It also includes different versions of the transport model, allowing the user to have a more or less detailed calculation of route allocation, network intensity and congestion.

4.4. Bridging the science — policy gap

The science-policy interface is one of the crucial elements in any DSS design and development process and creates the link between task 1 — scoping (mainly driven by the users) and tasks 2 and 3 — model selection and integration (mainly driven by the developers). Often research models are not directly suitable for incorporation in DSS (Oxley et al., 2004; Engelen, 2000). To move beyond a research model and provide added value to decision and policy-making, a model needs to connect to the policy context and process and, moreover, provide added value to those working with it. Section 4.4.1 describes the required system adaptations related to the science-policy interface and Section 4.4.2 focuses on the user-developer interaction process.

4.4.1. Translating scientific knowledge into policy-relevant information

In the process of linking scientific knowledge and policy-relevant information it is important to clarify the terminology used and to develop a common terminology for use among all involved parties. Very often problems occur because of a lack of understanding of one another’s vernacular. Discussing the details of what is meant and including a glossary in the scoping paper facilitates communication.

Research models often stop at the point where the process is modelled correctly for the purpose of deriving a specific scientific answer. This answer might have implications for policy questions (such as simulations of the Intergovernmental Panel on Climate Change groups), but the model itself does not facilitate an interactive use. For a DSS, however, it is crucial that the user can analyse the impact of various policy alternatives on a selected set of policy relevant indicators (see Volk et al., 2007, 2008). Furthermore, it is important that they can assess the sensitivity or robustness of these policy alternatives under different assumptions about the external driving forces that they cannot influence. This indicates that the user should have the option to set up coherent scenarios that consist of (a combination of) external factors and policy options, and to assess their impacts. These scenarios should then be used as drivers in the (scientific) models. The outcome of the assessment has to be provided as results that have meaning beyond the model itself. Therefore it is not sufficient to merely provide model output; it is crucial to define indicators that relate to the policy context. Very often there are discrepancies between the information needs from the users and the available data and models from the scientists. A scientific model can provide information about the dry-matter biomass of lemons, while a policy maker is interested in the yield, or moreover the profit in the lemon sector (example from the MedAction PSS). In such cases connections have to be created from the model results to the relevant indicators providing feedback to steps 2 and 3 of the methodology.

4.4.2. The role of social learning in the design and development process

One of the most important current challenges in the design, development and implementation of DSS is to ensure that they will actually be used by the intended users. In bridging the science — policy gap, users and developers need to work together towards shared goals as active co-producers in the social process of knowledge construction. Often problems arise in the early phases of the design and development process in understanding each other: because using integrated models for policy support is not (yet) common practice, users often do not know what they can expect and what the limitations of such systems are. Developers normally have a scientific or technical background and often have no experience in policy making, resulting in a limited understanding of the policy practice and the process in which policy decisions are being made and to which the DSS aims to provide support. To be able to learn from one another, openness plays a crucial role. Developers need to manage the expectations of users by communicating both the potential and the limitations of DSS to avoid unpleasant surprises during the implementation and use of the systems. Users need to explain their daily practice, the policy process they are involved in and the organisational structure and context they operate in. As described by Oxley et al. (2004) this process of social learning requires building relationships of mutual trust and respect. In doing so, it is essential that contact is frequent, personal and relaxed. Unfortunately, time for interaction is often limited due to its high costs. During the development of the DSS described in Section 2, the frequency of personal interaction between users and developers varied between one and six times per year and the duration of the interaction between half a day and a few weeks.

Developers are often focused on explaining the contents of the system, more than its use value, while the interest of the users is often in the latter. Using real cases on how the system can provide support to actual policy issues and the policy process is beneficial in assisting users to better understand what a DSS can and cannot be used for, because they directly see the benefits and limitations of the approach for their daily practice. This helps them to provide better direction to the further development because of their improved understanding of the (potential) usefulness and usability of the system. Real cases are best sought as a collaborative effort of the champions, providing a direct link to the users, and the architect, who forms a direct link to the development team, to ensure they are relevant and feasible.

In practice, many development trajectories that include users focus on the factual elements of policy making: policy options, indicators, etc. as described in Section 4.2, while the institutional, cultural, and personal issues are often forgotten or neglected. Like any decision-making process DSS development as well as the decision to use a DSS depends on individual differences and includes human emotion and imagination. The individual plays a central role as creator, actor and carrier in the decision-making...
process and organisational decision processes are often driven by the forces of affect, insight and inspiration of these decision makers acting collectively (Langley et al., 1995). This makes the role of champions crucial in the design and development process and especially during the implementation. Our experience teaches us that champions seem to be personally inspired, and their actions in turn inspire behaviour of others. Although the position of the champions in their organisations varies amongst the different example DSS in this paper, their common characteristics are that they are visionary people with a good network within their organisation (and often beyond) and a high interest in exploring new techniques, the latter facilitating the communication with the developers and in particular the architect(s).

4.5. Developing a usable and user-friendly interface

An important task for the developer of a DSS is to bridge the gap from scientific tools to user friendly systems, by creating a graphical user interface (GUI) that is easy to use, provides access to different policy options and external factors and visualises model output and indicators. Because the DSS described in this methodology encompass complex integrated models and aim to provide policy support, the GUI should be able to provide access to two different types of users: the policy makers or their resource people who use the system as part of their policy process and who carry out impact assessment studies with the model; and the scientists or modellers who can update the underlying data and parameters and possibly even the model structure. The first group benefits from a GUI that follows the steps of a scenario or impact assessment process. The second prefers to look at the system components and values easy access to individual disciplinary models.

It is very important to keep the different goals of the two types of users in mind. In most cases, exposing the full flexibility that is offered by the models adds no value to a policy user. On the contrary, it will actively hinder optimal use by them. Providing them with a list of configuration parameters and leaving it up to them to decide the best combination for their specific use detracts from the system’s usability. Design of user-friendly interfaces is about anticipating the user’s needs, adopting their worldview and expressing the workings of real world processes in a way that fits their experiences, needs, vocabulary and expectations — and requiring no more user action than strictly necessary.

The LUMOCAP, WISE, Xplorah and DeSurvey DSS provide examples of a dual user interface according to the division described above: a policy interface and a modeller interface (Fig. 4). Only those settings that could in the real world be influenced by policy makers were included in the policy interface. The sub-models with all their adjustments are still accessible by scientific users through the modeller interface. The elements of this part of the user interface are grouped per model; each individual model has its own access point through the system diagram (similar to the conceptual diagram described in Section 4.1). Access to settings for the policy user is structured not by where they fit into the model, but according to their logical function. On a high level, access is organized by the steps that a user takes to carry out an impact assessment analysis: configure drivers, create integrated scenarios, run the simulation, review output through the indicators and do comparative analysis. Zooming in on those parts, settings and outputs are grouped together by their type and their domain; for example all economic policy measures together, all external factors together, all ecological indicators together, etc. An overview of the policy and modeller interface of the LUMOCAP PSS is provided on the LUMOCAP website (JRC, 2009).

4.6. Implementation

There are some aspects related to the implementation of DSS that need to be taken into account and discussed during the design and development process. These aspects relate to:

- Decisions about the organisation and persons who are going to use the system (see also Sections 4.1 and 4.4.2). Many organisations are still organised in a very sectoral way and integration often takes place at a rather high political level. This poses questions as to where in the organisation such systems can best be operated and by whom. When organisations are open to make substantial changes in the workflow to operate in a more integrated way, the likelihood of success to implement the DSS enlarges. In such a process transition management techniques can be very beneficial (Weick and Quinn, 1999; Walker et al., 2007).
- The fact that many people resist change, and the experience that time available to learn and explore the potential of a DSS is often limited, threatens the use and implementation of DSS. Since the use of DSS in planning and policy making is still very novel, it is expected that implementation will become easier once a critical mass of users has been reached. According to Moore (1991) this would require a move from the early adopters of the systems (the technology enthusiasts and visionaries) to the early majority (the pragmatists). At the moment, we see a very important factor of success in having one or more champions at different levels of the organisation (Moore’s early adopters). Getting key people on board and involving them throughout the development process enables

![Fig. 4](image)

Fig. 4. Dual user interface of the Xplorah DSS. (a) (Left) shows the steps of a policy impact assessment on the left hand side and provides details about drivers, the creation of scenarios, indicators and analysis capability on the right hand side. (b) (Right) shows the systems diagram which gives access to the data and parameters of the underlying models by clicking on the name of the model component.
a user-oriented development process in which others feel comfortable to contribute at later stages. All systems we have developed and that are being used at present have such a person in their organisation. It should be noted that the presence of a champion in the early phases of the development and implementation is no guarantee for long-term use. Dependence of one person is a risk factor that should be avoided if possible or at least mitigated at the earliest opportunity. Embedding the system in a wider group of people in user organisation(s) during the implementation phase is therefore of crucial importance to secure a long-term use.

4.7 Use and maintenance

The above-mentioned tasks all have the aim to provide a useful and user-friendly DSS. The real test of the system is however this task: Is the system able to provide added value to policy development and analysis? And are the intended users able to work with the DSS? The more the different prototypes have been used, tested and improved to support policy cases (see task 4), the higher the chances of success in this step. Although it is expected that the DSS can be used for several years to support one or several organisations, this does not mean that no enhancements or updates to the system are required. Maintenance of the system is unavoidable. Systems need to be updated with recent data and adapted to be able to support new policy requests. Not all organisations are willing to work with the system themselves and prefer consultants to update the system, carry out the assessment work or facilitate workshop sessions. Discussions about maintenance and support during the development process, including the availability of the accompanying human and financial resources, are crucial.

4.8 Evaluation of the DSS

As can be seen from Fig. 3, evaluation of the DSS should take place after different tasks in the development process have been completed and provides the main feedback to make adaptations to earlier tasks. Ideally a first evaluation takes place after the design phase in which conceptual solutions for tasks 1–5 have been agreed upon. The next evaluation rounds take place after the development of each prototype, and final evaluation takes place when the system is being used. This latter evaluation includes not only the evaluation of the system for the current practice, but also the updates required to meet future expectations. Evaluation of the DSS and the included integrated model with all its components is not an easy task. Diez and McIntosh (2009), Van Delden (2009) and Volk et al. (in press) have provided a set of criteria to evaluate these types of systems. Volk et al. (in press) have used these criteria to evaluate several existing systems and Huser et al. (2009) and Van Delden et al. (2008b) have used the criteria presented in Van Delden (2009) to evaluate WISE and MedAction. All authors agree that the success of DSS and their use in policy or planning organisations depends on a combination of hard, technical aspects and soft, contextual aspects.

This section will focus on the three main groups of criteria and base this on the guidelines provided by Parker et al. (2002) for situations where formal analysis and testing of a model may be difficult or unfeasible. They state that “the essential, contemporary questions one would like to have answered when seeking to evaluate a model (are):

1. Has the model been constructed of approved materials, i.e., approved constituent hypotheses in scientific terms?

2. Does the behaviour approximate well that observed in respect of the real thing?

3. Does it work i.e. does it fulfill its designated task, or serve its intended purposes?”

Evaluating the behaviour of an integrated model is a very time consuming task, since most steps need to be carried out for each individual model, for sub-sets of models within the integrated model and for the integrated model itself to obtain a good understanding of the results and to be able to improve the structure and assumptions in an appropriate way. The evaluation procedure for integrated models is similar to the procedure carried out for the evaluation of individual models, described in several papers (STOWA/RIZA, 1999; Seppelt, 2003; Jakeman et al., 2006), with the notion that special attention should be devoted to the boundary conditions of the models, the feedbacks between the models, the integration of the different spatial and temporal resolutions and the (scientific) validity of the integration. In brief, the steps to evaluate a model’s behaviour include:

- Calibration over a historic period to test if the model is able to simulate past developments and to estimate and adjust model parameters and constants to improve the agreement between model output and a data set (Rykiel, 1996).
- Validation and/or evaluation over a different dataset to test if the structure and parameter settings found in the calibration perform well for a period beyond the calibration period (Walker et al., 2003; Van der Sluijs et al., 2005; Jakeman et al., 2006). This step is aimed at testing if the model is able to perform well for a period beyond the calibration period itself.
- Sensitivity analysis to explore the sources of variation that influence model output (Norton, 2008; Saltelli et al., 2000; Ziehn and Tomlin, 2009). Changes made in the input parameters are related to changes found in the output of the model. There are many methods of sensitivity analyses including one-at-a-time and two-at-a-time techniques (see Saltelli et al., 2000). In the former only one model parameter at a time is changed and compared with the model output. In a multivariate sensitivity analysis, multiple parameters are changed systematically and compared with model output (Kocabas and Dragicevic, 2006).
- Robustness checks to test the impact on the extreme values for the main drivers.
- Characterisation and analysis of data and model uncertainty (Walker et al., 2003; Van der Sluijs et al., 2005; Jakeman et al., 2006). Uncertainty must be considered in any modelling exercise, but is especially important in integrated modelling because of the risk of error propagation. When using integrated models for policy support acknowledging and communicating that model results are uncertain is essential. As Funtowicz and Ravetz (1990) state: “high quality (scientific information) does not require the elimination of uncertainty, but rather its effective management.”

Since integrated models comprise several individual models, it is no surprise that a complete data set for calibration and validation purposes for the entire model is often not available. Therefore pragmatic choices have to be made and sometimes the behaviour of the entire system has to be tested based on data sets for a few individual models, assuming that if the integrated model is able to provide realistic results for these data sets, it also has to provide realistic results for the other components, because of its integrated nature. The lack of data availability – and data quality - also puts emphasis on internal validation to check if the processes are
modelling correctly (part of the first question) and expert judgement for evaluating the long-term behaviour of the system.

For evaluating DSS the questions posed by Parker et al. (2002) should be seen in a broader perspective. It is not sufficient that the DSS meets the scientific standards; it should also adhere to the principles of legitimacy throughout the development process, i.e. are the appropriate stakeholders included? Are design decisions made in a proper way? Is the development process transparent? This means that compared to an evaluation of a model, where it is possible to evaluate the model's structure and the result, there is much more focus on the process.

Credibility should also be looked at in a larger context. Where scientifically validated models contribute to the credibility of the system to users, scientific validation is by no means the sole factor. Including users throughout the development process, discussing the underlying assumptions, and transparency with respect to the models included, greatly contributes to this as well.

The third question posed is closely related to the concept of saliency that states that research outputs must be seen by stakeholders as relevant to their decision-making process. This criterion is a crucial one for DSS and relates to the ability to bridge the gap between the scientific models and the information relevant for the user. One can speak of successfully bridging the science-policy gap when a DSS is used in the (daily) practice of the policy analysts or policy makers (including their resource people). Many DSS fail because the focus is too much on developing a scientifically correct model and too little on providing added value to the policy context and process. Saliency can be enhanced by applying the system to actual policy cases, to see to what extent the DSS is able to provide added value and what its limitations are. This evaluation relates to the usefulness of the system (as defined in Section 4.1) as well as its usability (as defined in Section 4.5). To avoid a mismatch between science and policy-relevance during the final evaluation, it is best carried out during each iteration loop, and especially when the system is in its conceptual phase(s) (see also Section 4.4.2 on social learning).

5. Discussion

5.1. Involvement of users

Throughout the entire design and development process user interaction is of crucial importance; not only to ensure that their input is included in the further development, but also because including them enables social learning on the side of the users as well as on the developers' side (scientists and IT-specialists). It is unrealistic to demand from users a detailed specification document at the beginning of the design and development process, simply because they are not aware of what can be expected and what limitations have to be taken into account.

Although required, iterations are time-consuming and can therefore frustrate some users involved, because they feel that work is repeated over and over again. An opinion often heard from the user groups is that they feel used, irrelevant or ignored because they participate in the design and development process, but do not hear back what has been done with their ideas and do not see them incorporated in the final product. Discussing with users the time line of the development process and providing feedback on the way their input is used, is therefore crucial in building trust and ongoing commitment. The Creating Futures project is an excellent example of successful communication between users and developers. Through workshop sessions and questionnaires, user input is collected and later on feedback is provided on the project website (Creating Futures, 2009; Huser et al., 2009). The LUMOCAP project has maintained a 'strategy paper' throughout the duration of the design and development process describing the goals, the current status, the next steps forward and the way (user) recommendations are taken into account.

5.2. Cost efficiency of an iterative process

Throughout the paper we have argued the need for an iterative development. Nonetheless, one should not underestimate the time and related costs for these iterations (e.g. Jakeman and Letcher, 2003). While some iterations are crucial to establish social learning, it is important to reflect on the efficiency and effectiveness of revising earlier decisions in each iteration round. Resources are limited and priorities have to be defined at early stages. Financial consequences of making substantial changes to earlier versions are high and should therefore be weighed against the benefits of those changes. Setting these priorities and creating efficient and effective iteration rounds is still a topic for further research. A thorough evaluation after the design phase in which conceptual solutions for tasks 1–5 are found can avoid major revisions during later stages after (more expensive) software implementation has been undertaken.

5.3. Hard versus soft factors: what is important at which stage of the process?

In the introduction of this paper we have stressed the importance of both hard and soft factors in the design and development process. Although both play a role throughout the process, their importance for the actual uptake and adoption of the systems by users varies throughout the process. Through experience we have learnt that in the early stages very often developers are selected based on their earlier experience and the faith users have that the developer team is able to deliver a useful and user-friendly DSS. Once a selection for a certain developer team is made and a first prototype has been developed, there often is a commitment from the user side to further improve this prototype rather than to investigate the possibility for other available prototypes. In this phase the trust that is built up in the team of users and developers is more important than the actual quality of the system. It is only after users start to use the system and validate its results that quality becomes a much more important issue. Also at this stage, users very often do not intend to look around for a system that might fulfill their needs better, but will focus on improving the system that they have adopted and have commitment for. To some extent this course of events is logical and even desirable, since it allows the build up of knowledge about a certain system in the organisation. On the other hand, it can also be a problem if a certain developer team (and related approach) is selected very early in the process because users might limit themselves to an approach that does not suit their needs best; at that stage they do not yet have the ability to be able to make such a fully informed decision.

5.4. Transferability of DSS

The focus of this paper is on the design and development of DSS. However, since DSS development is a very time-consuming and correspondingly expensive process, reusing parts of existing systems in new applications is highly beneficial. A major question that arises is to what extent systems are reusable? Can the software remain the same and can the adaptations required to fit the new context or region be carried out solely by changing data and parameter values? Or are adaptations to the underlying models or user interface required? Having a software environment in which model components from different model developers can be easily plugged in and connected is seen as highly desirable by many users.
and developers, and frameworks that focus on modularity and reusability of components are garnering attention. The examples provided in this paper are developed with one software platform (Geonamica) and many of the model components are reused in several applications all around the world. It is our experience that developing integrated models from existing model components that are already in the software platform takes anywhere from several weeks to several months. Although time-consuming, this is short compared to the time and effort required to develop a new model. We have also learnt to work with rather small components (even within models in the same discipline) as this makes the reusability much simpler; one can suffice with replacing those components that need to be changed. The more often we apply a similar configuration of models to a new region, the quicker it goes, because we start to develop a better feeling for the integrated model, start to develop a calibration procedure for it and can learn from the version previously applied to other regions.

6. Conclusions

In this paper we have proposed an approach to the design and development of DSS, which provides a methodological basis for the requirements discussed by Diez and McIntosh (2009), Van Delden (2009) and Volk et al. (in press). The proposed methodology emphasizes an iterative design and development process that enables social learning of the different groups involved. We have distinguished several important elements for models that aim to provide support to policy processes, and to which we refer as DSS in this paper:

- Synthesis of relevant drivers, processes and characteristics of the real world system at relevant spatial and temporal scales.
- An integrated approach linking economic, environmental and social domains.
- Connection to the policy context, interest groups and end-users.
- Engagement with the policy process.
- Ability to provide added value to the current decision-making practice.

The methodology to design and develop these systems includes a close interaction between the different parties involved (end-users, scientists, IT-specialists and architect). We believe that this iterative development enhances the usefulness and usability of the DSS and leads to an improved uptake of the system in user organisations.

In current DSS development the interaction between the architect and the champion(s) plays a crucial role throughout the design, development and implementation phase. The champion is the direct link to the users, the architect to the development team. To succeed, the champion and the architect should gain the respect of the users and the development team, respectively. They should develop trust and reliance with one other, and communicate effectively.

It is our strong belief that DSS such as the ones described in this paper, and their development processes, are a beneficial and often essential mechanism to resolve the complex problems that confront society in areas like catchment management, regional development, spatial planning, coastal zone management or global change. They can provide substantial added value in assessing the relative impact of a range of policy alternatives under various external conditions, and thus support robust policy development and analysis. Nonetheless there are a number of scientific, technical and use challenges that need to be faced for DSS to move from their current pioneering use to widespread mainstream use in administrations and their supporting organisations.

There is a movement within the research and development community to provide more sophisticated DSS. However, we must walk before we can run. We need to concentrate on achieving a sufficient and useful integration of these systems more routinely. Otherwise DSS development will remain an academic exercise. More successful practical examples will help in gaining a wider understanding and acceptance of these systems and improve the efficiency and effectiveness of the associated iterative development process. Shifting the focus of the development from the contents of policy-making to its context and process will in our opinion enhance the use of these systems, as will an improved understanding of the organisational culture and structure. Because using integrated models for policy support is still very novel and most organisations are organised by sector, making use of transition management techniques to implement these systems could be very beneficial.

We believe that the above-mentioned actions will enhance the use of DSS. To be able to provide good quality results we feel that the following points require more attention: data quality and consistency between different data sources, integration of models operating on different scales and developed using different modelling paradigms and a stronger focus on calibration, validation, sensitivity analysis and uncertainty analysis.

Finally, we feel that the current direction of software development for DSS, focusing on modularity and hence small, flexible components, provides a good basis for reusability of components and thereby more efficient applications to different regions and policy contexts. We caution however that good software solutions do not solve all integration problems: science, IT and use have to go hand in hand to make DSS a success.

Finally, the methodology proposed should not be viewed as a harness, but rather as guidance for structuring the process of developing a useful and user-friendly DSS with attendant high scientific and software standards. As always the availability of data, time, financial and resource constraints have to be taken into account. This process is very often a balancing act in which acceptable solutions for all should be found, without compromising on aspects that are of crucial importance to the different interest groups.

Acknowledgments

The New Zealand Foundation for Research, Science, and Technology funds the Creating Futures project (ENW0601), the European Commission has funded the development of LUMOCAP (FP-6 SSPE-CT-2005-006556), MedAction (FP-5 EVK2-2000-22032) and DeSurvey (FP-6 IP 003950), the Puerto Rican government funds the development and implementation of the Xplorah SDSS and the German government finances the Elbe-DSS.

References


