18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Web-based management of simulation models - concepts, technologies and the users' needds

Haubrock, S.N. [1], **F. Theisselmann** [1,2], **H. Rotzoll**[1] **and D. Dransch** [1]

[1] *Helmholtz Centre Potsdam GFZ German Research Centre for Geosciences*
[2] *Humboldt-Universität zu Berlin, Department of Computer Science, Systems Analysis*
Email: soerenh@gfz-potsdam.de

Abstract: Simulation models are commonly developed by scientists to evaluate scenarios that show potential developments of the Earth system in the past, present and future. In order to provide adequate information systems that facilitate access to simulation models to a broad, heterogeneous user community, the deployment of such models on web servers provides a technical fundament. However, many aspects need to be taken into account for setting up operational, user-friendly web-based systems that include access and administration tools for simulation models. Data integration, data exchange, scenario management, and visualisation are among the most important functionalities to be accounted for, while usability needs to be aimed at by choosing an appropriate abstraction level and providing a careful interface design. Usually simulation models encapsulate complex algorithms, which have been developed by domain experts and implemented based on very diverse technologies. In order to provide the functionality of such models to users over the Internet, standardisations such as the Web Processing Service developed by the Open Geospatial Consortium (OGC) help to specify the technological framework, but do not provide concepts for guaranteeing the aforementioned functionalities and usability. An additional requirement from the administrator and developer perspective is to offer a minimum level of flexibility in information architectures in order to adapt and exchange single components such as a simulator or data base. In many cases, this flexibility stands in conflict with a rapid, use-case specific development.

In this paper, different integration concepts for hydrological simulation models into web-based management systems are compared to each other. All concepts were developed to fulfil the requirements of heterogeneous user groups, ranging from scientists to re-insurance companies. Their implementation in prototypical real-world systems was performed in inter-disciplinary groups of experts in Hydrology and Information Technology. While the first three integration concepts focus specifically on functionality (legacy model encapsulation, integration of real-time data, scenario management) and usability (user interface, visualisation) for single simulation models or static process chains, the fourth use case outlines a way towards more generic service composition based on a workflow management system. A comparison of the potential and limitations of these architectures results in a discussion of aspects to be taken into account for making simulation models accessible and usable for science, industry and governmental agencies. From our experience of designing, setting up and running the developed systems we conclude that functionality and usability are in the main interest of the end users of such systems. Each user group has different requirements, depending on their expertise and objectives. However, a clear, easy-to-use user interface is far less error-prone and avoids semantic problems for lay users, while experts require complex control mechanisms to run, calibrate or even re-design their modelling infrastructure. Integrating third-party data sources is possible, but requires well-defined machine readable user interfaces. For system administration and sustainability, system architectures incorporating a higher flexibility and implementation effort in the setup phase are seen to pay off in the long term. It is very important that all relevant aspects have to be specified in the design phase of a web-based management system for simulation models. Depending on this specification, the target system focuses either on implementation speed or flexibility, which comes with the cost of a more complex service-based infrastructure. The demand for using and accessing simulation models has increased in number and complexity in recent years. With the availability of appropriate concepts and technologies from information technology, integrating such systems into a web environment is a worthwhile, yet challenging task for the modelling and Information Technology communities.

Keywords: Web-based management, Decision Support Systems (DSS), Simulation model access, Distributed system.

1. INTRODUCTION

With the success of simulation models in generating insights into real-world systems and allowing for calculating synthetic scenarios for the past, present and future, their users have spread from the academic field towards policy makers, local authorities and the economy. Each user group has different objectives for using such models, such as decision making or improving eco-system understanding. A single underlying modelling system can potentially be used to fulfil these needs. However, different views on the model and is results need to be provided to the users in order to deliver the information in an optimum way. Decision-support systems (DSS) have been developed to target a specific group of users. While DSSs comprise several other functionalities as well, simulation models often form their functional core. The concept behind a DSS has not changed significantly over the years. However, recent technological progress allows for more sophisticated architectures and the integration of additional functionalities and other components, but these new powerful technologies may be relatively complex.

Thus, a recent trend in software development is to abstract from use-case specific implementations to more generic approaches that are adapted to each use case by configuration rather than implementing the overall system from scratch. This trend has been reflected by the integration of geographical information systems and simulation models (e.g. Schowanek et al., 2001), the development of modelling frameworks (e.g. Kralisch et al., 2005; Kralisch et al., 2007), as well as recent standardisation developments in the domain of geomatics, such as the Web Processing Service (WPS) (Schur, 2007). At the same time, service-oriented architectures (SOA) have been established in information technology, but have been marginally used in the modelling communities so far. The EU project ORCHESTRA reflects the requirements for service infrastructures in the disaster management field by focusing on interoperability among risk management authorities in Europe (Denzer et al., 2005).

Basic recurring functionalities of such software systems comprise the simulation itself, storage and exchange of data between model components, as well as the presentation of the simulation results to the user. Parallel access by multiple users or repeated access to stored scenario data requires the management of sessions within a sophisticated information system. User groups may additionally be interested in systematically analysing scenario ensembles, in performing sensitivity analyses on the models and in comparing results from different modelling approaches visually and statistically. For this purpose, statistical packages and visualisation frameworks need to be provided. For realizing multi-user access, the management of scenarios needs to be extended by queuing and scheduling components. Also, the trend to deploy sensors in the natural environment has increased the amount and availability of data collections significantly. An automatic integration of sensor data in simulation systems would certainly benefit the users of simulation models. Each of these functionalities poses an additional challenge and increases complexity for a simulation model management system.

Over the years, distributed research groups have developed their own simulation models independent from one another. Input data and results are commonly stored in local files or databases, geographical information systems or web mapping services are used for the visualisation of results. For two reasons, the local deployment of modelling components in monolithic systems or modelling frameworks does not meet the requirements for adequate information systems. First, users are generally not provided access to the models independent of their location and computer system configuration in a 24/7 manner. Second, the laborious integration of legacy or third-party components needs to be performed for each stand-alone system individually, while modularising the component would be a singular task.

In order to overcome these limitations, web-based distributed information systems are a frequently discussed concept. The usage of a common web browser in combination with a basic user authorisation and authentication system is one practical solution to use models independent of time, spatial or system configuration constraints. Web portals function as central access points for all resources and for managing the user accounts. The distributed components that comprise the information system consist of simulation models, data sources, mapping services and middleware systems that manage scenario queuing and scheduling tasks in a multi-user environment. However, the application of modern web-based technologies for simulation modelling comes at the cost of using occasionally complex technologies with particular characteristics.

In this paper, we present four different use cases of web-based management systems for simulation models and compare their characteristics with respect to usability, flexibility, functionality, and implementation effort. For the purpose of software system characterisation, we refer to usability as the subsumption of different aspects according to Nielsen (1993). These aspects include efficiency, learnability, memorability, error rates and satisfaction, affecting the user's ability to understand and use the software interface and its output. Thus, usability needs to be guaranteed by choosing an appropriate abstraction level from the complexity of both, the simulation model itself as well as the technological system in the background, depending on the expertise of the user groups. Providing specific views on the simulation model, combined with a careful interface design is necessary to find a compromise between maximum functionality and usability.

Each use case focuses on one of the different aspects, legacy model integration, usability, linkage of different simulation models and flexibility. They were all implemented to an operational or prototypical status and are in parts accessible to registered users of the Natural Disasters Networking Platform (NaDiNe) (Haubrock et al., 2006).

2. TOWARDS A WEB ARCHITECTURE FOR SIMULATION MODEL MANAGEMENT

The use of the World Wide Web has undergone a paradigm shift recently with the establishment of stateful interactive web applications in the context of the so-called Web 2.0. Applications are increasingly deployed on the Internet, with access being made available either via web pages (for human users) or web services (for machine use). The concept of encapsulating application components as web services resulted in the formation of service-oriented architectures (SOAs), in which services communicate with one another via standardised protocols (e.g. SOAP messages), which are most often sent on top of the Hypertext Transfer Protocol (HTTP). Corresponding security mechanisms are available on this basis to guarantee encryption and authentication, such as Secure Sockets Layer (SSL) or Web Service Security.

Recent development in web technologies and service-oriented computing has resulted in a well-defined set of concepts and standardised implementations, making the development of distributed applications feasible. Standardisations have additionally triggered the spread of web services within the geomatics community. The Web Processing Service (WPS) developed by the Open Geospatial Consortium (OGC) is an example of such standards. It defines how processes (usually with a spatial context) can be accessed and used in a syntactically standardised way. The concept behind WPS and other web services supports the loose coupling of components such as client interface, database and core simulation service that is the concrete implementation of these services is hidden behind specified interfaces.

Wrapping simulation models in such a way allows for the integration of legacy software as executable software components, but often makes some adaptation of the interfaces necessary. With this architecture, an exchange or update of a component on the server is independent of the user interface and therefore in most cases a smooth process as long as the interfaces are not changed. Conceptually, each task that needs to be performed in a simulation system (data input, storage, processing, and visualisation) can be described as a single service and made accessible by implementing or encapsulating it as a web service.

The major aim of providing web-based management facilities for simulation models is to optimise the aspects of functionality, usability and access by utilizing state-of-the-art technologies. The complexity of the simulation model and its algorithms can be encapsulated, while control needs to be guaranteed by providing appropriate, user-specific interfaces. Moreover, legacy simulation models that have often been developed by domain experts and implemented based on diverse technologies need to be integrated without loosing controllability.

Web-based access to the information system is realized using web browsers as the sole client software. User-friendly graphical user interfaces (GUIs) provide access to simulation models for users as well as for developers and administrators who may wish to control the models with different or additional functionality provided via the interfaces. In the information systems that are presented in this paper, we integrated several off-the-shelf third-party frameworks providing web-based mapping and diagram functionalities. Storage of geographical data in spatial databases allows for connection of standardised Web Map Services (WMS) (de la Beaujardiere, 2006) or commercial systems such as Google Maps. In our examples, all spatial data visualisation is based on Geoserver with PostGIS as the storage system. Background data is integrated from Google Maps using the Google API. User interfaces make u se of asynchronous communication with AJAX technology.

While the real-world scenarios sketched in the following all have different purposes and target heterogeneous user groups, they share the common architectural principle based on this service-oriented paradigm.

3. ARCHITECTURES OF WEB-BASED SIMULATION MODEL MANAGEMENT

3.1. Use case I: providing access to legacy models

The Flood Loss Estimation Model (FLEMO) (Thieken et al., 2007), a legacy simulation model assessing damages caused by inundation in German river catchments, has been integrated into a Web environment. For web integration, it was compiled to an executable file without further adaptation of the model. The executable can be controlled via command line or from within an application based on its Application Programming Interface (API). Data input is realized using parameters and data input files, while scenario metadata and the output of the simulation are stored in a database. The latter functionality has been added explicitly for the web integration. The user interface of the web-based information system consists of a web form with an upload facility for the input file (a compressed raster layer of the inundated area). Results are integrated in a web page in form of a table and map view. Additionally, the results (Shapefiles and ASCII tables) can be downloaded for local post processing. Figure 1 sketches the overall architecture.

By means of a web-browser (User Interface) the user accesses a portal (Management). The portal is responsible for accessing models and data (Modelling, Spatial DB). One characteristic of this architecture is its relatively simple setup: All components are straightforward to implement, configure and deploy with relatively little additional effort. The fact that the simulation is integrated as an executable that can be easily triggered from within a software component allowed for its integration as an atomic process. Simulation identification numbers embedded in the URL and the serialisation of the results in a database allow for accessing the results at any time.

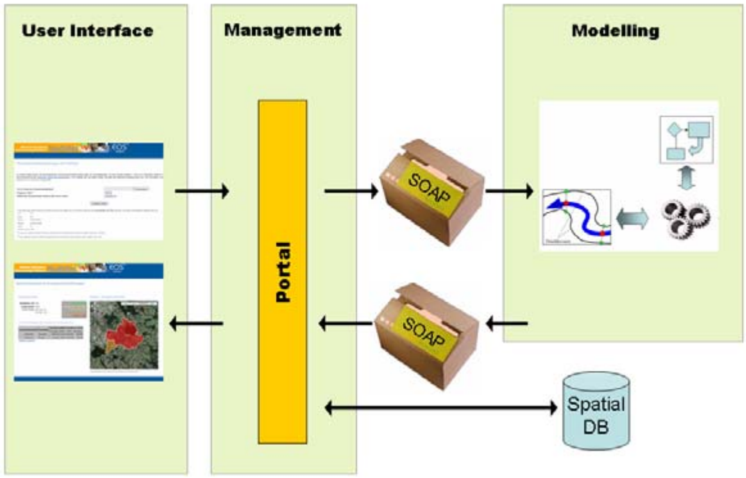

Figure 1. Architecture of direct legacy model integration into web environment.

3.2. Use case II: enhancing usability

Simulation model access and control functionality is highly dependent on the characteristics of the user groups. In this use case, a single dike breach simulation model (Vorogushyn et al., 2006) was made accessible to two different user groups (Figure 2): 1) a group of hydrologists interested in easily accessing the model with full functionality and integrated real-time gauge data and 2) lay users running basic scenarios with partly pre-defined input parameters and datasets (Haubrock et al., 2007).

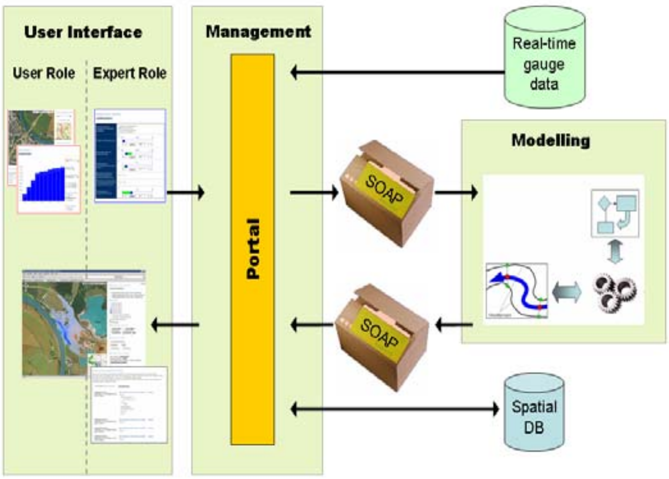

Figure 2. Architecture with user-specific model access and sensor data integration.

Different user interfaces were developed for each group that can be accessed via a web portal. For the lay users, intuitive parameterisation was aimed at by complementing direct number inputs with interactive graphical control elements, e.g. to localize dike breach in maps, or to specify water levels by manipulating time-series diagrams. For the expert user group, the number of input parameters and different control elements adjusting each parameter was higher compared to the lay users for a more detailed model parameterisation. Documentation of scenario results was realized in a hierarchical tree view, allowing both user groups to overview all simulation runs and to find all necessary information.

3.3. Use case III: linking simulation models

Within an interdisciplinary project, a prototypical integrated flood control centre was established in Germany, linking several simulation models and data sources as a distributed service-oriented architecture (SOA). This information system contains an evaluation service of remote sensing data for monitoring and detecting dike breaches along the Elbe River (triggering process), a dike breach model, and different damage assessment models of inundated areas as well as a visualisation service presenting the simulation results in an integrated way. The general concept was to install one web service for each component on different servers that are located at each providing institution, resulting in a distributed architecture.

This SOA-based information system realizes model coupling by orchestration of single web services within a static process chain that is defined within the server-component. Queuing and scheduling mechanisms are integrated for each simulation model component allowing for concurrent multi-user access.

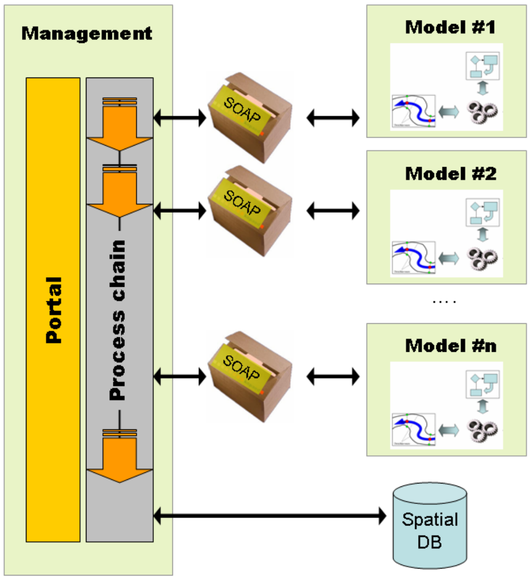

Figure 3. Server side of architecture with multiple simulation models and components linked in static process chains.

One of the disadvantages of this use case and the previous two is that components are hard-wired, thus, the substitution or update of components entails source code adaptation and makes this process cumbersome and error-prone. Additionally, triggering a simulation model was implemented by compiling it to an executable program (at design time) and setting up a process at the server operation system (at runtime). However, this approach lacks control functionality at runtime, such as querying the current simulation state and stopping or pausing a simulation. However, these functionalities cannot be provided without changing the simulation model source code.

3.4. Use case IV: providing flexibility

In contrast to the approaches described so far, the limitations of restricted flexibility of the workflow and the limited control of components are targeted in this use case, which has been developed in the context of the NaDiNe-Dike project (Theisselmann et al., 2006; Theisselmann et al., 2009). The information system provides information about possible dike breaches in the modelled area, including a respective simulation model. For this purpose, the management of a single simulation model and additional software components is sub-divided into two parts: a static component controlling the user interface and an additional middleware component in form of a workflow engine to be used for setting up individual process chains, i.e. sequential processes that are linked to each other by exchanging data and passing control. Each component (simulation, data access and storage, or mailing) is encapsulated as a separate web service and can be integrated into a workflow definition. Single components can be substituted separately while the system is online, provided that interface definitions remain unaffected, allowing for a straightforward adaptation of the overall architecture without rebuilding the application.

An additional gain in flexibility is reached by a more sophisticated service encapsulation of the simulation model. Instead of embedding an executable application file in a web service wrapper, the functionalities of initialising and iterating within a simulation process are implemented. This is realised by cross-compilation and makes manipulation of the model source code necessary. While the gain in control and functionality is large and often desired by the expert users of such models, its implementation is affected with significant additional effort. However, when simulation models are developed from scratch, the effort of implementing interfaces is likely to be easily manageable.

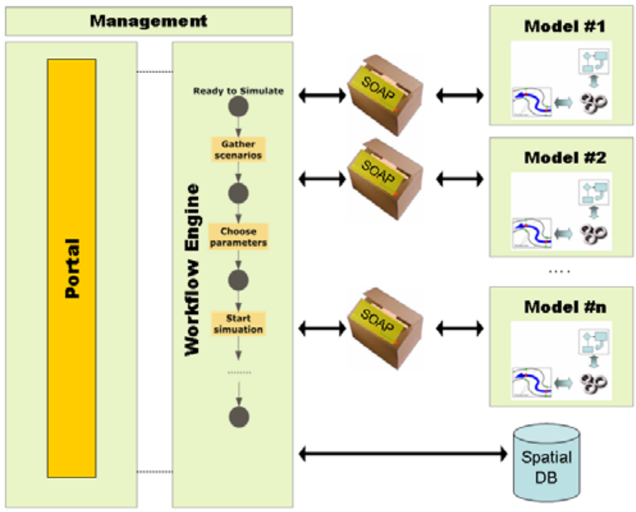

Figure 4. Server side of architecture with multiple simulation models defined in adaptable workflow.

4. DISCUSSION AND CONCLUSIONS

Four different real-world use cases for managing simulation models in a web environment have been demonstrated in this paper. Each use case was designed to best fit the heterogeneous user requirements. The complexity of the architecture increased throughout the use cases by adding varying functionality and usability. Table 1 summarizes the evaluation of the aspects that have been studied in this paper for each use case.

Table **1**. Comparison of design aspects for setting up web-based simulation model management systems.

Use case & aim	Usability	Flexibility	Functionalities	Costs and effort
Use case I: providing access to legacy models	• simple and clear web GUI • intuitive parameterisation	• single model exchangeable • static parameterisation	• simulation invocation • result visualisation • result download	• web page implementation • data base setup • model encapsulation
Use case II: enhancing usability and functionality	• user group specific GUIs	• single model exchangeable • user group specific parameterisation flexibility	• user group specific functionalities • sensor data integration	• as in Use case I + sophisticated web GUI (AJAX)
Use case III: linking simulation models	• specific perspectives for each model and overall process chain	• single models exchangeable • process chain flexibility after deployment	• single service calls • process chain execution	• as in Use case I + process chain implementation
Use case IV: providing flexibility	• high complexity caused by workflow definition • straightforward execution	• several components: high flexibility for re-use and service-chaining through web service approach	• all functionalities provided (data management, visualisation, model execution) • process chain setup	• high effort for architecture setup • high effort for adapting legacy models • low adaptation costs

In the process of setting up a web-based simulation model management system, these aspects can be taken into account for deciding which architecture fits best the individual needs of the users and administrators. Comparing the benefits and costs of each architecture, different perspectives need to be considered. From a user's point of view, functionality and usability are more important than the costs to setup a management system. From a provider or institutional perspective, sustainability and flexibility are most important. Sustainable implementations need to be maintained, changed and extended from time to time. With systems as implemented in the use cases I to III, the cost for each use case is relatively low, while re-implementing the system for an additional or adapted model is very cumbersome and ineffective. Consequently, the huge effort necessary for setting up a flexible management system in the beginning, consisting of a highly modular, service-based architecture with a workflow management system as middleware, pays off in the long term.

It has been shown in the use cases that many perspectives exist, making the definition, design and development of managing systems complex. For the scope of optimum implementation efficiency, usability and functionality, a separation of concerns seems necessary between those groups involved in the design of the models, which are generally developed by the scientists of a specific domain, and the technological development. In reality, this separation is not always accomplished, resulting in sub-optimal model implementations performed by non-experts in software development on one side, and consumption of resources that ideally focus on model improvement on the other. These issues have been recognised already, resulting in sophisticated modelling frameworks. The next logical step is to provide framework functionality and flexibility to the web, combined with a standards-based integration of data sources and third-party system components.

Encapsulating the complexity of implemented methods and algorithms is a necessary prerequisite to allow the target audience to make use of the underlying services. This aim can today be fulfilled with web-based architectures making use of state-of-the-art technologies and standards.

REFERENCES

de la Beaujardiere, J., (2006), OpenGIS Web Map Service v1.3.0 – Implementation Specification, Technical Report, OGC 06-042, http://www.opengeospatial.org/standards/wms

Denzer, R., Güttler, R., Schimak, G., Usländer, T., Atkinson, M., (2005), ORCHESTRA Development of an Open Architecture for Risk Management in Europe. In: Jakeman, A. and Swayne, D. A., (Eds.), Environmental Software Systems: Environmental Risk Assessment Systems: Sesimbra, Portugal, International Federation for Information Processing, 10pp.

Haubrock, S., Theisselmann, F., Dransch, D. (2007), Service-Oriented Architectures for Natural Disaster Management., In: Proceedings of the Joint CIG/ISPRS Conference on Geomatics Solutions for Disaster Management, CIG/ISPRS Joint Conference, Toronto, Conference CD.

Haubrock, S., Dransch, D., Kreibich, H., Merz, B., Plattner, S., Voigt, S., Mehl, H. (2006a), Natural Disasters Networking Platform (NaDiNe). In: Amman, W., J. Haig, C. M. S. Huovinen, (Eds.), Proceedings of the International Disaster Reduction Conference. Vol. 1. Swiss Federal Research Institute WSL, 74.

Kralisch, S., Krause, P., Fink, M., Fischer, C., Flügel, W.-A., (2007), Component based environmental modelling using the JAMS framework. In: Kulasiri, D., Oxley, L., (Eds.), MODSIM 2007 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2007, Conference CD.

Kralisch, S., Krause, P., David, O., (2005), Using the Object Modeling System for hydrological model development and application. *Advances in Geosciences*, 4, 75-81.

Nielsen, J., (1993), Usability Engineering, Morgan Kaufmann, 26-27.

Schowanek, D, Fox, K., Holt, M., Schroeder, F.R., Koch, V., Cassani, G., Matthies, M., Boeije, G., Vanrolleghem, P., Young, A., Morris, G., Gandolfi, C., Feijtel, T.C.J., (2001), GREAT-ER: a new tool for management and risk assessment of chemicals in river basins, Water Science and Technology, 43 (2): 179-185.

Schur, P., (2007), OpenGIS Web Processing Service v1.0.0 – Implementation Specification, Technical Report, OGC 05-007r7, http://www.opengeospatial.org/standards/wps.

Theisselmann, F., Haubrock, S., Dransch, D., (2009), Service-oriented Architecture for Environmental Modelling - The Case of a Distributed Dike Breach Information System. In: Kulasiri, D., Oxley, L., (Eds.), MODSIM 2009 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, July 2009, Conference CD.

Theisselmann, F., Haubrock, S., Vorogushyn, S., Dransch, D., (2006), NaDiNe-Deich - ein Geoinformationsservice für das Katastrophenmanagement, In: Clemen, Ch., (Eds.), Entwicklerforum Geoinformationstechnik 2006: Junge Wissenschaftler forschen., Shaker, 27-36.

Thieken, A.H., Olschewski, A., Kreibich, H., Kobsch, S., Merz, B. (2008), Development and evaluation of FLEMOps - a new Flood Loss Estimation MOdel for the private sector. In: D. Proverbs, C.A. Brebbia, E. Penning-Rowsell, (Eds.), Flood Recovery, Innovation and Response: 315-324.

Vorogushyn, S., Apel, H., Lindenschmidt, K.-E., Merz, B., (2006), Coupling 1D hydrodynamic, dike breach and inundation models for large-scale flood risk assessment along the Elbe River, *Geophysical Research Abstracts*, 8, 10565, EGU General Assembly, Wien, April 2006.