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

Coupling the RBIS Environmental Information System and the JAMS Modelling Framework

S. Kralisch [1], F. Zander [1], and P. Krause [1]

[1] *Department of Geoinformatics, Hydrology and Modelling, School of Chemical and Earth Sciences, Friedrich-Schiller-University, Jena, Germany*
Email: sven.kralisch@uni-jena.de

Abstract: The pressure on environmental systems is increasing worldwide due to population growth and the consequences of climate change. Adaptable tools and methods are needed to elaborate information, develop understanding and strategies for sustainable use and management of environmental systems. Such tools should assist scientists, natural resource managers and decision makers in their work by providing them with (1) sufficient information about relevant drivers, attributes and factors from measured data, (2) tools and methods for user friendly access and integrated analyses of such data, i.e. environmental information systems (EIS), and (3) tools for estimating additional information not available as measurements, i.e. environmental simulation models. For an integrated assessment of complex environmental systems and their related problems a seamless and preferably standardized coupling of environmental information systems with environmental modelling systems is desirable. On the other hand, both parts should always be useable as standalone versions to avoid over-complexity of toolsets which can hamper or even permit their use for projects which tackle less complex problems.

Data exchange between environmental information systems and simulation models are of increasing importance as the amount of available environmental information rises together with the complexity of conceptual simulation components and hardware capability. Due to their explicit representation of data interfaces, modelling frameworks are especially suited for direct interaction with EIS.

The research at the Department of Geoinformatics at the Friedrich-Schiller-University Jena is reflecting the needs described above with the development of a number of software tools. They provide services and assistance for specific spatio-temporal related environmental problems emerging from integrated research projects. In general, all developed tools (1) use open source software wherever possible to be cost effective, (2) are provided as open source software to others and (3) are highly flexible and adaptable to ensure useability for a wide range of environmental problems.

In this paper we briefly present two of our developed tools, namely the River Basin Information System (RBIS) and the Jena Adaptable Modelling System (JAMS) and show how these two systems can communicate to share data among each other. RBIS is a web-based information management system with a focus on time series and geospatial data. It provides user friendly services for data input and output, and an adaptable set of functions for data analysis, data management and data enrichment. JAMS is a modelling framework for the component based development and application of environmental simulation models. It was used to implement a number of process-oriented models mainly for simulating water and solute transport processes.

Keywords: *Modelling Frameworks, Water Resources Management, Environmental Information Systems*

1. INTRODUCTION

Integrated environmental resources research carried out by interdisciplinary research groups requires a comprehensive and holistic system analysis and process understanding (FLÜGEL 2007). It in turn relies on powerful and effective data management. It also depends on information sharing strategies based on best available knowledge to support researchers and decision makers. Considerable work has been carried out regarding open access database systems that provide environmental geo- and time series data to a broader research community via the internet. Examples for these systems are *Pangaea* (http://www.pangaea.de), *FAOSTAT* (http://faostat.fao.org) or *UNEP Grid* (http://www.grid.unep.ch) who provide access to a variety of geo-spatially related data. The literature review reveals that there are numerous attempts to organize and coordinate national, regional, and global geo-spatial data. Nevertheless, it can be stated that integrating complex information such as geodata, time series and meta-information together with statistical analysis and mapping capabilities in a fully integrated way is still challenging (PCGIAP 1998, BASSOLET 2000, WILLIAMSON et al. 2003, MANSOURIAN et al. 2006).

Many existing data management systems are web-based system where users are enabled to search, locate and access the provided data sets. Usually, a catalogue service is being implemented to handle search requests providing information about available data sets and how they can be accessed. Technically, data access is then granted through HTTP by using web services like the Open Geospatial Consortium’s (OGC) Web Feature Service (WFS), Web Coverage Service (WCS) for geodata or Sensor Web Enablement (SWE) Services for time series data (http://www.opengeospatial.org/standards). From a project management perspective, a deficit of such approaches is that users are not able to easily integrate own data, to alter existing data sets or to manage associated meta-information. This is rarely a problem of missing features in available standards, but a lack of support for these functions on the client’s side. Regarding the data management design, many of these systems also lack a fine-grained user permission management (e.g. write-permission to only a subset of available data) or map-based user interfaces, like e.g. map-based search functions and model-specific interfaces. Often, such systems show little flexibility for implementing user-required extensions and applications. On the other hand, non-web-based geodata information systems (e.g. the Chinese Cryospheric Information System, Li et al. 2003) are usually stand-alone solutions which do not support multi-client, distributed applications.

2. RBIS – AN ENVIRONMENTAL INFORMATION SYSTEM

The *River Basin Information System* (RBIS, Flügel 2007) developed at the Department of Geoinformatics, Hydrology and Modelling of the University of Jena addresses the abovementioned deficits. This modular-structured application consists of *(i)* a comprehensive and spatially enabled relational database and *(ii)* an application server providing a web-based user interface for easy data access. The common layout of RBIS follows a standard 3-tier architecture design (figure 1) using the *PostgreSQL* DBMS (http://www.postgresql.org) as a storage engine.

2.1. Data types

RBIS was developed with a scope on the following types of data:

1. Geodata (raster and vector formats, e.g. ESRI Shapefiles or GeoTIFF)
2. Time series data, e.g. climate or runoff data
3. Documents, e.g. common binary files
4. Simulation models, e.g. references to software, parameter values and input data required to reproduce simulation results

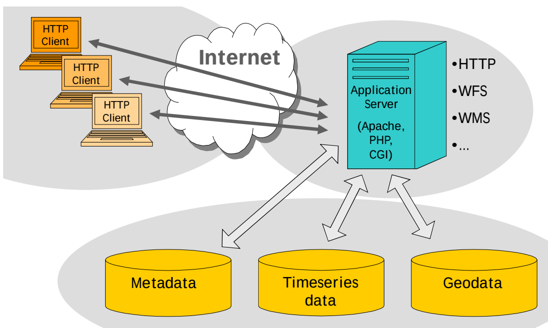

Figure 1: RBIS technical layout

Accordingly, the RBIS application features different software components for managing these data types, called *RBISmap* (geodata), *RBISts* (measuring stations and time series), *RBISdoc* (binary documents), and *RBISsim* (simulation models). In order to express relations between these information (e.g. between measuring stations and the associated measured time series data) the software components can flexibly be associated to each other by means of database relations.

2.2. Meta-information

An important aim during the RBIS development was the description of all stored data by comprehensive meta-information. As a starting point, the ISO 19115 metadata standard for geodata (ISO 2003) was used to describe all stored geodata. To provide information about other data types like measuring stations, time series data or binary-formatted documents the ISO 19115 data model was used where possible and extended where necessary. As an example, the ISO 19115 *Responsible Party* relation is used describing both: persons or organizations that are in charge of some geodata and those who are responsible for maintaining measuring stations. On the other hand, such stations can be further described by additional information e.g. regarding the parameters being measured or the year of establishment or closing.

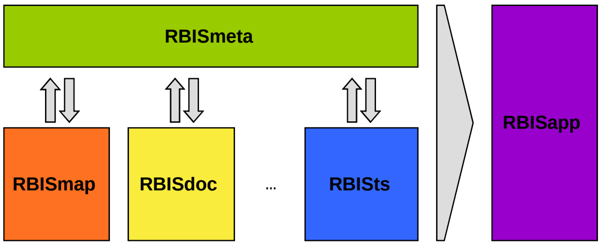

Figure 2: Interactions of RBIS software components

Metadata management is handled by the *RBISmeta* software component which associates meta-information datasets to the datasets of other components. Its interaction with other parts of RBIS is sketched in figure 2.

2.3. Software architecture

As a platform for geodata storage and visualization, the *MapServer* (http://www.mapserver.org) and *PostGIS* (http://postgis.refractions.net) software packages were used. In order to increase ease of use, some modifications to the MapServer where applied, e.g. a web-based map editor was added. The RBISts, RBISdoc and RBISsim components where developed from scratch using available software libraries whenever possible. As a web stack, a standard Linux environment with Apache web server, PHP and PostgreSQL/PostGIS (LAPP) was used. The whole system is built upon open source software, ensuring a cost-effective deployment and operation.

Data are stored within database tables where possible and in the file system otherwise (e.g. raw data). According to the different application components, three different databases (for geodata, time series data and meta-information) are used. This differentiation simplifies the system's administration and opens the opportunity for a distributed deployment.

Since RBIS is accessible via the web and designed for distributed access by users with different levels of access authorizations, a flexible and powerful permission management was an important precondition for its acceptance and trustful use. This is achieved by granting access rights to single users and user groups based on:

1. Data types, e.g. access to geodata only,
2. Dataset ownership, e.g. write access to a dataset only for its creator,
3. Dataset location, e.g. access to stations whose position are within some defined spatial vicinity, i.e. a polygon,
4. Dataset properties, e.g. access to time series data which use limitation attribute is set to 'public'.

In addition to the permission management, transparency is accomplished by means of a comprehensive access logging mechanism which is responsible to record write access to all data.

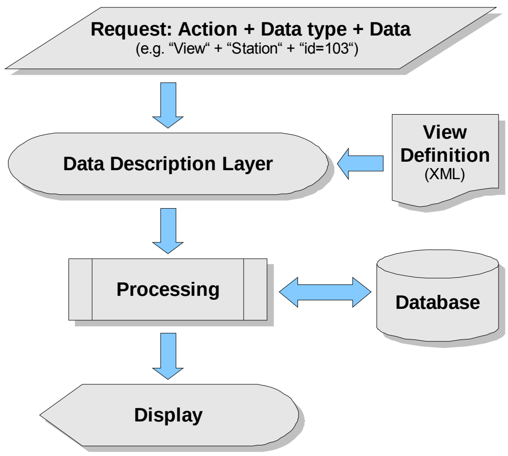

Figure 3: RBIS information access via data description layer

Due to its flexible and modular database and application design, RBIS can easily be tailored to meet emerging demands for new data types and relationships between them. This is achieved by a description layer in the RBIS software that takes care of read/write access to database contents. This layer takes an XML document as input which describes database relations and attributes that define a certain data type, e.g. measuring stations. The XML document is then being evaluated by RBIS in order to create SQL statements and user interfaces needed to perform different actions on these data, e.g. search, browse, edit, add or delete measuring station datasets (figure 3).

Taking benefits from this approach, all actions have to be implemented only once and can be used to access all kinds of RBIS data. New data types can easily be added by *(i)* adopting the database design (e.g. by adding new relations and attributes), and *(ii)* defining how RBIS shall access these data by means of a XML document.

2.4. Time series management

The RBISts functions for the management of time series data have been designed with special focus on the demands of environmental models. Besides functions for uploading, visualizing and downloading the following features regarding meta-information (e.g. about the associated station) and the time series itself are provided:

1. Data gap analysis of uploaded data and their detailed description,
2. Aggregation methods for creating various temporal resolutions,
3. Gap filling toolbox with various algorithms for reconstructing missing data.

Especially the last feature is useful when dealing with measurement data. An example is the *inverse distance weighting* method (Shepard 1968) to fill data gaps using data from neighbouring stations. This method is very common since every time series is associated with a measuring station and thus has a geographical location. The best suited filling method is selected by a rule-based algorithm whose rules can easily be defined by the user, e.g. inverse distance weighting for temperature data is used if at least 3 stations provide measured data for the time interval and are located in a range of less than 50km.

In addition to the system's methods for gap filling, RBISts also allows the upload of externally created gap data to complete a time series. In all cases, detailed meta-information about the used procedure and all relevant parameters are stored along with the gap data to make sure that any data modification is always transparent, reproducible and reversible.

3. COMPONENT-BASED MODELING WITH JAMS

3.1. Modelling frameworks

To tackle problems of integrated water resources management (IWRM), like e.g. raised by the implementation of the European Water Framework Directive, flexible software systems that can be used for the development and application of environmental simulation models are needed. These demands are covered by modelling frameworks which *(i)* define a software architecture (e.g. by means of interfaces and abstract classes) that specifies how modelling components inside the framework can interact with each other and *(ii)* can control the creation, linkage and execution of such components.

Environmental modelling frameworks which are suited to cover the special demands of environmental simulation models are usually characterized by the following properties:

- They provide special data types that describe the spatial and temporal domains.
- The control flow inside these frameworks is not fixed, but can be configured from outside, e.g. by means of special components.
- They provide functions for managing and manipulating environmental data, e.g. for reading and writing time series data or for unit conversion.

The *Jena Adaptable Modelling System* (JAMS) is a framework that features these properties (Kralisch and Krause 2006). JAMS has been developed with the main objective to create simulation models that can simulate environmental processes at discrete points in time and/or space. This approach is widely-used by many distributed hydrological models applied in current practice.

JAMS is being developed at the Department for Geoinformatics, Hydrology and Modelling of the Friedrich-Schiller-University Jena, Germany. The main objective of the JAMS development is not the coupling of whole environmental models but rather the model creation from well-defined, scientifically sound process simulation components. As an example, these components simulate interception, potential evapotranspiration or soil temperature, each of these processes according to various algorithms. Depending on the model's purpose and available input data, a user can choose from the available process implementations that best matches her or his demands. Using JAMS, scientists can implement their expert knowledge as simulation components, even if little information about the later application context might be known at this stage (e.g. other components they might be coupled or used with). While providing a rich set of features useful for the development and application of simulation models (e.g. saving a models state or providing server based and parallel execution environments), the framework's invasiveness is minimized as much as possible in order to ensure a fast learning process. Process components implemented for JAMS and models constructed from them are focused on the hydrological and nutrient cycle (Kralisch et al., 2007).

JAMS is licensed under the GNU Lesser General Public License (LGPL) and can be downloaded from http://jams.uni-jena.de. The package includes the JAMS core system, its GUI extensions for model creation and execution, a library with frequently used components (e.g. for creating plots and maps from simulation results), and the J2000 hydrological modelling components. In order to test JAMS, the package also provides a preconfigured, ready-to-run J2000 model with test data.

3.2. JAMS model concepts

JAMS provides two specific types of building blocks for the creation of simulation models, named *JAMSComponents* and *JAMSContexts*. JAMSComponents can process input data (e.g. time- and space-variant environmental information) and calculate respective output data according to their implemented algorithm. This algorithm can range from environmental process representations (e.g. simulation of snow melt) to simple data processing tasks like spatial or temporal aggregation of calculated results or data I/O to the file system. Using JAMSComponents, an environmental process can be implemented without any knowledge about its later execution context, e.g. the temporal resolution or the type of spatial discretisation of the modelled area. The only precondition for the later application of a component in conjunction with others is the proper declaration of its desired input and provided output data by means of metadata in the component's source code.

JAMSContexts are special JAMSComponents working as containers for others, managing the control flow of their "child" components (e.g. iterative or conditional) and their data exchange with others. A JAMSContext can be seen as a scope that defines an environment for the execution of other software components. Most importantly, JAMSContexts are used to manage the repeated simulations of environmental processes in time and space.

Figure 4 shows an example of a typical model layout with different contexts and tasks executed within them. JAMSContexts are shown as grey, JAMSComponents as white boxes. Within the outermost *Model Context*, basic input data are read, e.g. spatial modelling entities. Its child components are called only once. The *Temporal Context* is used to read time-variant datasets once at a time, e.g. climate information, calling its children once for every simulated time step. Finally, the innermost *Spatial Context* contains environmental process simulation components. According to their position within the model, they are executed once for every time step and every spatial model entity.

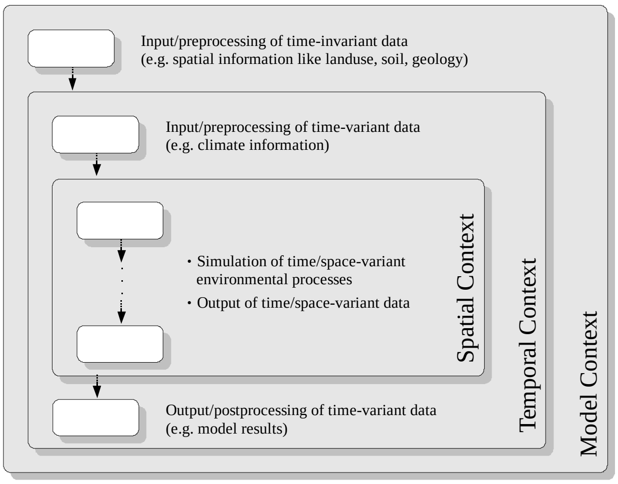

Figure 4: Typical layout of distributed environmental simulation models in JAMS

3.3. JAMS Data I/O

Data I/O in JAMS can be handled in two different ways. One possibility is that a JAMSComponent can be used to read or write data from any data source, e.g. files or databases. In this case, the component developer has to take care of reading the necessary information via the component's input data (e.g. file names), access the data source, and provide potential results (e.g. data read from a file) as output data. An advantage of this approach is a high flexibility in accessing external information. On the others side, such components usually need to be adapted to certain models and data formats which limits their reusability.

The second way to handle data exchange in JAMS is the usage of so-called *DataStores*. A DataStore in JAMS is an interface that can be used to read/write data from/to an external storage device, which might be a file as well as a database system or a network socket. It allows reading and writing of data tuples consisting of a number of data objects that are described by metadata, e.g. type, boundaries, and meaning. Figure 5 gives an overview of the JAMS DataStore architecture, showing two types of DataStores for reading (*InputDataStore*) and writing (*OutputDataStore*) data. In the case of InputDataStores, methods can be used to ask for (*hasNext* method) and retrieve (*getNext* method) available data tuples (*DataSet*). Accordingly, Output-DataStores provide methods to write data to the underlying storage device.

The data types currently supported range from numerical types over strings and calendar dates to spatial objects. In order to represent the latter, JAMS makes use of *Simple Feature* geometries as defined by the Open GIS Consortium (OGC, http://www.opengeospatial.org).

To access data from a JAMS DataStore, its ultimate data source and the software to access it needs to be identified and parameterized. This is done by the use of XML-based documents containing information in the *JAMS Data Description Language* (JDDL). Each JDDL document defines one DataStore and serves three purposes: (1) identify and parameterize an I/O software component that can access a certain type of data source, (2) define one or more data sources to be accessed by that component, (3) and in the case of an InputDataStore, define how retrieved data will be formatted and arranged.

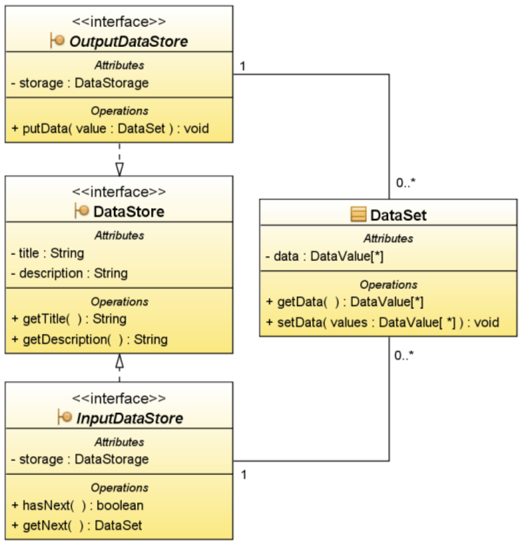

Figure 5: JAMS DataStore classes

The I/O software components can be provided by the user and made available to JAMS via a plug-in mechanism. As an example, this could be a piece of code which is able to read ASCII-formatted files containing time series information. Parameters could be the file name and the buffer size to be used.

The JDDL documents are provided as input for the JAMS model and are managed by the framework's runtime system. In order to access them from a JAMSComponent (e.g. for reading input data), it can ask the framework for a certain DataStore, using its title as an identifier. If existent, the framework will setup and initialize the DataStore and return it to the JAMSComponent, taking care of data access, connection sharing and error handling.

Comparing the two approaches to access data from a JAMS model, the DataStore shows significant advantages over a component's direct access to data sources. These are

- Centralized management and responsibility for data I/O,
- Data access definition by means of external, reusable XML documents,
- Data access sharing managed by the JAMS runtime system, and
- Outsourcing of data access functionality from JAMSComponents.

In addition, JAMS provides a caching mechanism for InputDataStores. When applying it, all data that are read from such a DataStore are locally cached. On demand, these cached data can later be used by the same DataStore as input in a fully transparent manner, e.g. in the case of aborted database access caused by instable internet connection or data that are not permanently available.

4. RBIS - JAMS INTERACTION

Given the RBIS data management functions and the JAMS data I/O approach described above, a coupling of both systems could easily be established. An I/O software component was implemented in order to access time series stored in RBIS. On demand, various meta-information about the time series can be made available to the DataStore if needed for further processing, e.g. location and elevation of measuring stations. The JDDL document is used to identify time series and associated meta-information and to format the data appropriately. Figure 6 shows how a JAMSComponent can access such a DataStore in order to read input data or to write simulation results.

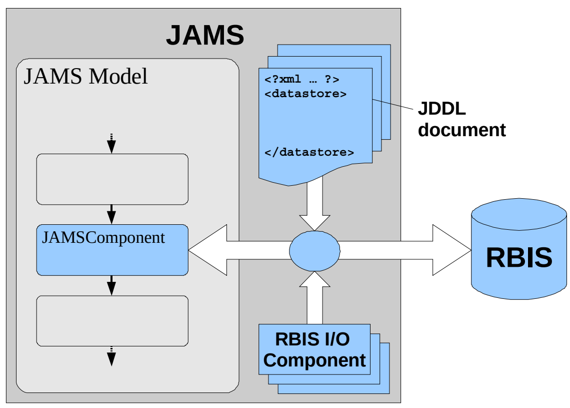

Figure 6: JAMS/RBIS data exchange

To support users in creating the JDDL documents for InputDataStores, RBIS was extended by a software wizard that helps in selecting the desired parameter, time interval, and temporal resolution, as well as in choosing a number of time series from a list of matching ones.

The resulting JDDL document can be downloaded from RBIS and made available to JAMS in order to read the chosen data. From a user's perspective, no additional effort is needed to prepare data, like gap filling, data formatting or checking for correct start and end dates in the case of time series data.

5. SUMMARY AND CONCLUSION

We have presented an environmental information system and a modelling framework as important prerequisites for integrated assessment of natural systems. First, we have introduced the River Basin Information System (RBIS) as an application for web-based management of geo-related data with a strong focus on meta-information. In addition, RBIS is especially suitable to manage time series data and provides various functions for analysing and filling data gaps, which usually is an important pre-processing step for environmental model applications. To use these functions, the Jena Adaptable Modelling System (JAMS) as a framework for environmental model development and application was extended by generic interfaces for data exchange. We demonstrated how these interfaces work and how they can be used to access data managed by RBIS.

Future work on the JAMS-RBIS interface will focus on further enhancing the output of simulation results to RBIS. Especially the automated creation of model application descriptions in RBISsim and the output of geodata layers representing simulation results are in the focus of ongoing development. These efforts will be further steps towards a true integrated modelling system.

The systematic coupling of environmental data management systems and modelling frameworks as introduced in this paper simplyfies analysis and processing of available information and supports researchers, decision makers and stakeholders in making best use of available knowledge. Therefore, we consider it is an essential prerequisite for integrated assessment of environmental systems.

ACKNOWLEDGMENTS

We would like to thank the German Federal Ministry of Education and Research for funding the JAMS and RBIS development in the course of the program "Entrepreneurial Regions – the BMBF Innovation Initiative for the New German Länder".

REFERENCES

Bassolet, A. (2000), An Overview of SDI Activity in Africa. 4th GSDI Conference, Cape Town, South Africa.

Flügel, W.-A. (2007), The Adaptive Integrated Data Information System (AIDIS) for global water research. Water Resources Management (WARM) Journal, 21, 199-210

ISO, INTERNATIONAL STANDARDS ORGANIZATION (2003): International Standard ISO 19115 Geographic information – Metadata. Reference Number ISO19115:2003(E).

Kralisch, S. and Krause, P. (2006), JAMS - A Framework for Natural Resource Model Development and Application. In A. Voinov, A. Jakeman, and A.E. Rizzoli, editors, *Proceedings of the iEMSs Third Biannual Meeting "Summit on Environmental Modeling and Software"*, Burlington, USA, July 2006. International Environmental Modeling and Software Society. ISBN 1-4243-0852-6.

Kralisch, S., Krause, P., Fink, M., Fischer, C. and Flügel, W.-A. (2007), Component based environmental modelling using the JAMS framework. In Don Kulasiri and Les Oxley, editors, *MODSIM 2007 International Congress on Modelling and Simulation*, Christchurch, New Zealand, December 2007. Modelling and Simulation Society of Australia and New Zealand. ISBN 978-0-9758400-4-7.

Li, X., Cheng, G.D. and Wu, Q.B. (2003), Modeling Chinese cryospheric change by using GIS technology. Cold Regions Science and Technology 36 (1-3): 1-9.

Mansourian, A., Rajabifard, A., Valadan Zoej, M.J. and Williamson, I.P. (2006), Using SDI and Web-Based System to Facilitate Disaster Management. Journal of Computers & Geosciences 32: 303-315.

PCGIAP, Permanent Committee on GIS Infrastructure for Asia and the Pacific (1998): A spatial data infrastructure for the Asian and Pacific region. PCGIAP publication, No. 1, Canberra, Australia.

Shepard, D. (1968): A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 ACM National Conference*: 517–524.

Williamson, I.P., Rajabifard, A. and Feeney, M. (2003), Developing Spatial Data Infrastructures: From Concept to Reality. Taylor & Francis.