International Environmental Modelling and Software Society (iEMSs)
7th Intl. Congress on Env. Modelling and Software
San Diego, California, USA, Daniel P. Ames, Nigel W. T. Quinn, Andrea E. Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2014-proceedings

Linking water resource network models to an open data management platform

P. Meier [a] **S. Knox**[a] **and J. J. Harou**[a]

[a] *University of Manchester, School of Mechanical, Aerospace and Civil Engineering, Manchester, UK (philipp.meier@manchester.ac.uk, stephen.knox@manchester.ac.uk, julien.harou@manchester.ac.uk)*

Abstract: The development of models for complex water resources systems frequently involves customising model formulations to fit unique institutional or physical problems. Linking existing model components is part of this process. As an alternative to pre-packaged models or building models based on components that comply to the same standard interface we present a model platform where network resource models (water, energy, transport, etc.) are linked to a data manager in a generic way. A modular architecture is achieved using a server for data storage. Clients connecting to the server, called Apps, provide import and export functions from and to various data formats. An App can also run models based on data retrieved from the server. As every component of an integrated model interacts only with the server, each component can be replaced easily. Model results and underlying input data are stored in one consistent database. This ensures that model results can be stored efficiently and reused in future modelling or analysis. The software hopes to allow researchers, consultants and decision-makers to rely on a common data management platform for the modelling, the planning and management of resource networks.

Keywords: model platform, water management models, integrated modelling, open source

1 INTRODUCTION

In order to adapt to the needs of decision-makers, resource network planning tools are evolving and a multitude of models exist in different areas (water, energy, logistics, etc.). Many models designed by researchers or consultants do not provide a user interface, instead model input data and control variables are organised in files. In order to change input data these files need to be modified. This process requires detailed knowledge about the file format used, can be error-prone and is time-consuming. Managing input files and model results can be a demanding task for large resource systems. These reasons explain why analysts and decision-makers often rely on software products that feature tools for organising and managing data [Harou et al., 2010].

For some modelling tasks it is necessary to chain together multiple individual model components. This off-line coupling approach is for example used to assess river basin management options under stable hydrological conditions, or to increase numerical stability of models for large river basins [Bravo et al., 2012]. When chaining different models, results from one model need to be transferred to the model next in line. Because each of these models has different requirements on data, transferring data from one model to the other can be difficult. Appropriate tools that store data in one single place can help managing model data and communicating results. This can help bridge the gap between the needs of researchers and practitioners which can slow knowledge transfer [Acreman, 2005].

In this article we propose a model platform for managing resource network data. This model platform aims at providing flexibility for managing large network datasets, while storing all data in a common database and relying on a generic user interface for data management and visualisation. The article is structured as follows: Different existing approaches to model building and data management in water resources management are discussed in Section 2. In Section 3 a new model platform for network based models is presented. This model platform presented is discussed and conclusions are drawn.

2 BUILDING MODELS FOR WATER RESOURCES MANAGEMENT

2.1 Integrated modelling

Creating an integrated model is based on exchanging data between different model components. There are two different approaches of integrating two or more models:

On-line coupling: Data is exchanged between models at run-time. At each time step data is exchanged between different model components. Feedbacks between different processes can be simulated.

Off-line coupling: Two or more models are run sequentially. One model provides input data for the model next in line. Feedbacks between separately modelled processes can not be considered.

For both integration approaches, data exchange between individual models needs to be enabled. This data exchange can be done through a standard interface (e.g. OpenMI). If a model does not comply to a standard interface, output data from the first model stored in files needs to be reformatted into input files needed by the next model. This reformatting is usually done using custom scripts or custom built computer programs.

2.2 Available tools

There are three main approaches to providing integrated models to end users without requiring them to implement data exchange between models using custom computer programs: Modelling shells, modelling frameworks, and model platforms (Figure 1).

Modelling shells provide a suite of prepackaged model components and a user interface that allows a user to develop an integrated model [Loucks and van Beek, 2005]. *Modelling frameworks* provide a common interface model developers can use for coupling their model to other models that comply with the same standard. *Model platforms* provide a common user interface for building the model of a physical system. Simulations or optimisations are carried out by separate programs that connect to the same database [Harou et al., 2010].

2.3 Modelling shells

Most interactive modelling tools available for water resources management belong to this category. Examples for such tools are WEAP [Johnson et al., 1995], AQUATOOL [Andreu et al., 1996], MODSIM-DSS [Fredericks et al., 1998], RiverWare [Zagona et al., 2001], MIKE BASIN [Jha and Gupta, 2003], MULINO [Mysiak et al., 2005], RIBASIM and HEC-ResSim.

Each modelling shell provides a finite set of models that are tightly linked to the user interface. Processes that are modelled are usually assigned to single nodes or links. Water flow through the network is calculated based on the mass balance of each node. Model components are coupled at run-time (on-line coupling).

Modelling shells can be used to develop integrated models of river basins in an intuitive and interactive way through a specialised customised graphical user interface and data storage system. River basins

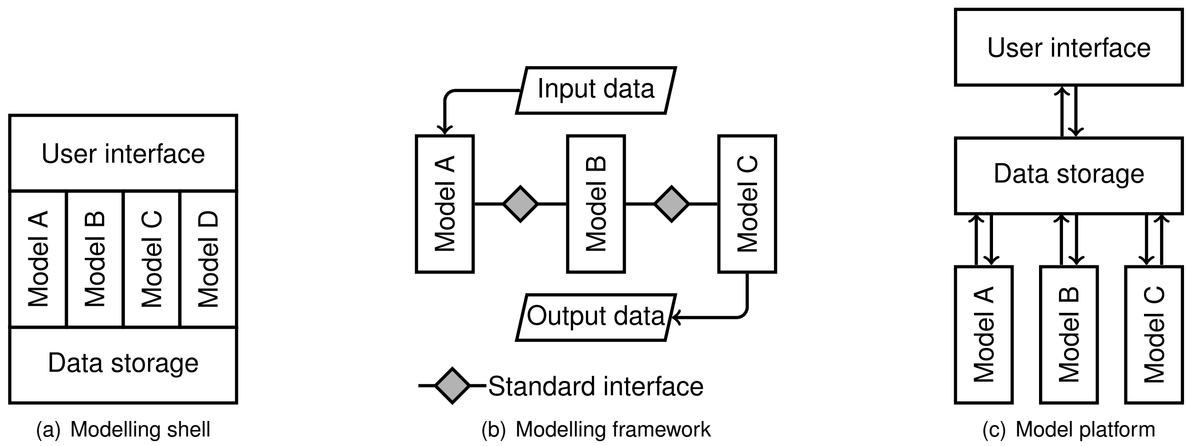

Figure 1. Integrated modelling approaches.

are represented as a network of nodes and links. Data, such as time series and parameters, are associated to nodes and links. All data is stored in a common database. Results of the simulation and intermediate results can be accessed and visualised at any time. Most modelling shells provide functionality for data import and export from and to external file formats.

2.4 Modelling frameworks

In modelling frameworks, models communicate through a standardised interface. This facilitates off-line and on-line coupling. Only models that implement this standard interface can be used. This means that either modifications on the model code itself might be necessary or at least using wrapper [Castronova and Goodall, 2010]. A wrapper is an independent piece of software that writes inputs to and reads outputs from a model while communicating with the standard interface.

Examples of modelling frameworks, are the Modular Modelling System (MMS) [Leavesley et al., 1996], the Object Modelling system (OMS) [David et al., 2002] and OpenMI [Gregersen et al., 2007]. MMS requires models to be built into a specialised model library. Existing models can be made compatible with OMS and OpenMI by adding a communication layer directly inside the model code or as a wrapper. Other model integration approaches such as the Simple Model Wrapper (SMW) [Castronova and Goodall, 2010], which is based on OpenMI and web services based models [Goodall et al., 2011] can also be classified as modelling frameworks.

Data exchanged between models is not written to files but transferred directly through the interface. This makes model coupling very efficient. For this reason modelling frameworks are mostly used to couple different models at run-time (on-line coupling).

2.5 Model platforms

Model platforms aim at providing flexibility for choosing models and at the same time store all data in one database. Data exchange between models is implemented by making all model components access the same database. One example of a model platform is *Delft-FEWS* from Deltares, a model platform for hydrologic modelling and early warning. A proof-of-concept of a generic model platform for water resources networks was presented by Harou et al. [2010]. This model platform, called HydroPlatform, provided a network builder and a common data storage, as well as export functions to several models. There are also approaches that combine different architectures. Leavesley et al. [1996] describe a system where an integrated model developed using MMS is coupled to a water management model into what they call a *database centered DSS*.

Model platforms can also provide functionality for chaining different models (off-line coupling). They are not suited for coupling different models at run-time. Data persistence and providence are promoted through storage in one consistent database.

Besides a standard data storage, model platforms feature a user interface that supports data management and lets the user carry out import and export of data. Models are loosely coupled to a model platform. Independent programs export data from the database to the format needed by a model and import model results from the files produced by the model. Due to the fact that each model component uses the same data storage, single components of an integrated model are exchangeable. The only condition is that a program that implements data exchange between the data store and the model is available.

3 PROPOSED MODEL PLATFORM

We present a new open-source model platform for network-based data management named *Hydra Platform*. The software is built around a server, to which a graphical user interface and Appss connect as clients. The server provides database access, and provides a programming interface based on a standard Web Service, through which Apps exchange data with the server. The server is implemented in Python and connects to the database using an ORM (Object Relational Mapper), which allows the server to connect to different database implementations. The server is designed such that App developers and users do not need to interact with the database. Instead the Web Service provides all functionality. This design also ensures that the server can be developed further without affecting the ability of Apps to access server functionality.

Around this core, Apps provide different functionality, such as data import and export from and to different file formats or connections to simulation and optimisation models. A graphical user interface (GUI) is a special case of an App, communicating through exactly the same programming interface as other Apps.

The main advantage of such an approach is its flexibility. Apps can be written in any programming language. The software can be run in a standard desktop environment as well as in a client - server, distributed setup. The possibility of installing the server on a remote machine anticipates potential enhancements such as web based user interfaces and cloud or cluster computing based services.

3.1 Networks and Scenarios

Network topology and associated data are stored independently in the database. A network is defined by a combination of nodes and links. Each network, node, or link has attributes appropriate to the model(s) that will be used with the network. Data is assigned to attributes through a scenario. Each network can have multiple scenarios. The types of data supported are descriptors (free form text), scalars, arrays and time series. An attribute on any node or link can be associated to data of any of these types. Descriptors accommodate non-numeric values or contain scripts for linking rules or constraints to nodes and links. Numbers can be entered as scalars, arrays and time series.

Within a scenario, groups of nodes and links can be defined. These groups reflect interdependencies a model relies on to perform certain types of calculations. A group of nodes can determine whether a specific process is modelled on those nodes or not, regardless of their type. In a water resources system, groups can be used for example to distinguish nodes that feature a storage from those which do not. The group then determines which kind of mass balance equation is used for a specific node.

3.2 Apps

Apps are adaptors between the server and various external models or data formats. They communicate with the server through a standard Web Service interface (currently SOAP). An App is usually a wrapper around a model or a simple export and import function for the file format read by the model. In most cases developing an App only involves data manipulation. No knowledge of model algorithms is needed for the development of an App. However, there is no restriction on what an App should look like, as long as it connects to the server. Since every App is a stand-alone piece of software, it can be implemented using every programming language that supports the programming interface. Also, there are no restrictions on licensing.

The graphical user interface (GUI) is a special case of an App, which allows the user of Hydra Platform to conveniently interact with the data storage. The GUI provides facilities for visually creating and managing even complex networks, building scenarios and manage data. Extensive data visualisation tools for decision support systems are provided. All requests from the GUI to the server will go through exactly the same interface used for Apps.

3.3 Templates

Each model has its own requirement of what attributes a node and a link can have. If a network is to be used with a specific model, the attributes on network elements need to follow the structure needed by the model. The set of attributes on a node or link specifies its *type*. All node and link types needed for a model are specified by a *template*. A template is an XML file that contains a list of attributes for each node and link type. Types specify the kind (scalar, array, time series, descriptor) and valid ranges of data expected for an attribute and whether the attribute is treated as a variable. A variable is an attribute which acts as a place holder for expected results of a model. Every App that connects to a model needs to be accompanied by a template.

In Hydra Platform, users may want a single network to be shared with several models. In this case, a single network can be exported to multiple different data formats for use with different models. To achieve this, a consistent view of node and link attributes must be used by all Model Apps. This is achieved using a controlled vocabulary which provides guidelines for naming attributes. Such a controlled vocabulary can be agreed on within a modelling community.

Templates in conjunction with a controlled vocabulary provide the core functionality needed for sharing data between different models (Apps). Templates facilitate the integration of different models by providing a simple way of running them on the same consistent dataset.

4 CONCLUSIONS

Model platforms provide a flexible tool for building and managing water resources models. This paper describes the design and characteristics of a generic open model platform called Hydra Platform. This model platform can be used to manage data for network based models. Network topology and datasets are stored independently, which allows linking multiple variations of data (scenarios) to one network. A programming interface allows Apps and a user interface to access this database. Apps import and export a networks topology and datasets to the desired format. Chaining models is done by Apps importing from one format and exporting to another. To make multi-model interaction consistent, Apps agree on what the data stored in the common database looks like. This is achieved by templates, which provide the information on what node or link types are needed for a specific App, and which attributes they need to have. Individual components of an integrated model can be exchanged without affecting other model components because it only interacts with the server.

The proposed system provides a platform to build and maintain complex decision support models for network resource systems including water resources. A common user interface allows researchers, consultants and decision-makers to use a familiar graphical environment for data management and visualisation that is independent from the models applied.

ACKNOWLEDGEMENTS

The project is partially funded by the UK government through the Technology Strategy Board (TSB).

REFERENCES

Acreman, M. (2005). Linking science and decision-making: features and experience from environmental river flow setting. *Environmental Modelling & Software*, 20(2):99 – 109. Policies and Tools for Sustainable Water Management in the European Union.

Andreu, J., Capilla, J., and Sanchs, E. (1996). Aquatool, a generalized decision-support system for water-resources planning and operational management. *Journal of Hydrology*, 177(34):269 – 291. Decision-Support Systems.

Bravo, J., Allasia, D., Paz, A., Collischonn, W., and Tucci, C. (2012). Coupled hydrologic-hydraulic modeling of the upper paraguay river basin. *Journal of Hydrologic Engineering*, 17(5):635–646.

Castronova, A. M. and Goodall, J. L. (2010). A generic approach for developing process-level hydrologic modeling components. *Environmental Modelling & Software*, 25(7):819 – 825.

David, O., Markstrom, S. L., Rojas, K. W., Ahuja, L. R., and Schneider, I. W. (2002). *Agricultural system models in field research and technology transfer*, chapter The object modeling system, pages 317–331. Lewis Publishers, Boca Raton, FL.

Fredericks, J. W., Labadie, J. W., and Altenhofen, J. M. (1998). Decision support system for conjunctive stream-aquifer management. *J. Water Res. Pl.-ASCE*, 124(2):69–78.

Goodall, J. L., Robinson, B. F., and Castronova, A. M. (2011). Modeling water resource systems using a service-oriented computing paradigm. *Environmental Modelling & Software*, 26(5):573 – 582.

Gregersen, J. B., Gijsbers, P. J. A., and Westen, S. J. P. (2007). Openmi: Open modelling interface. *Journal of Hydroinformatics*, 9(3):175 – 191.

Harou, J. J., Pinte, D., Tilmant, A., Rosenberg, D. E., Rheinheimer, D. E., Hansen, K., Reed, P. M., Reynaud, A., Medellin-Azuara, J., Pulido-Velazquez, M., Matrosov, E., Padula, S., and Zhu, T. (2010). An open-source model platform for water management that links models to a generic user-interface and data-manager. In Swayne, D. A., Yang, W., Voinov, A. A., Rizzoli, A., and Filatova, T., editors, *2010 International Congress on Environmental Modelling and Software Modelling for Environments Sake, Fifth Biennial Meeting, Ottawa, Canada*. International Environmental Modelling and Software Society (iEMSs).

Jha, M. K. and Gupta, A. D. (2003). Application of mike basin for water management strategies in a watershed. *Water International*, 28(1):27–35.

Johnson, W., Williams, Q., and Kirshen, P. (1995). Weap: A comprehensive and integrated model of supply and demand. In *Proceedings of the 1995 Georgia Water Resources Conference, April 11 and 12, 1995, Athens, Georgia.* Georgia Institute of Technology, Carl Vinson Institute of Government.

Leavesley, G., Markstrom, S., Brewer, M., and Viger, R. (1996). The modular modeling system (mms) – the physical process modeling component of a database-centered decision support system for water and power management. In *Clean Water: Factors that Influence Its Availability, Quality and Its Use*, pages 303–311. Springer Netherlands.

Loucks, D. P. and van Beek, E. (2005). *Water Resources Systems Planning and Management - An Introduction to Methods, Models and Applications*. UNESCO PUBLISHING.

Mysiak, J., Giupponi, C., and Rosato, P. (2005). Towards the development of a decision support system for water resource management. *Environmental Modelling & Software*, 20(2):203 – 214. Policies and Tools for Sustainable Water Management in the European Union.

Zagona, E. A., Fulp, T. J., Shane, R., Magee, T., and Goranflo, H. M. (2001). Riverware: A generalized tool for complex reservoir system modeling. *JAWRA Journal of the American Water Resources Association*, 37(4):913–929.