



Jul 1st, 12:00 AM

# An open-source model platform for water management that links models to a generic user-interface and data-manager

Julien J. Harou

Didrik Pinte

Amaury Tilmant

David E. Rosenberg

David E. Rheinheimer

*See next page for additional authors*

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

Harou, Julien J.; Pinte, Didrik; Tilmant, Amaury; Rosenberg, David E.; Rheinheimer, David E.; Hansen, Kristiana; Reed, Patrick M.; and Reynaud, Arnaud, "An open-source model platform for water management that links models to a generic user-interface and data-manager" (2010). *International Congress on Environmental Modelling and Software*. 510.  
<https://scholarsarchive.byu.edu/iemssconference/2010/all/510>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact [scholarsarchive@byu.edu](mailto:scholarsarchive@byu.edu), [ellen\\_amatangelo@byu.edu](mailto:ellen_amatangelo@byu.edu).

---

**Presenter/Author Information**

Julien J. Harou, Didrik Pinte, Amaury Tilmant, David E. Rosenberg, David E. Rheinheimer, Kristiana Hansen, Patrick M. Reed, and Arnaud Reynaud

# **An open-source model platform for water management that links models to a generic user-interface and data-manager**

**Julien J. Harou<sup>1</sup>, Didrik Pinte<sup>2</sup>, Amaury Tilmant<sup>3</sup>, David E. Rosenberg<sup>4</sup>, David E. Rheinheimer<sup>5</sup>, Kristiana Hansen<sup>6</sup>, Patrick M. Reed<sup>7</sup>, Arnaud Reynaud<sup>8</sup>, Josue Medellin-Azuara<sup>5</sup>, Manuel Pulido-Velazquez<sup>9</sup>, Evgenii Matrosov<sup>1</sup>, Silvia Padula<sup>1</sup>, Tingju Zhu<sup>10</sup>**

*1 University College London (UK) ([j.harou@ucl.ac.uk](mailto:j.harou@ucl.ac.uk)), 2 Enthought (BELGIUM), 3 IHE Delft (Holland) & ETH (Switzerland), 4 Utah State University (USA), 5 University of California, Davis (USA), 6 University of Wyoming (USA), 7 Pennsylvania State University (USA), 8 Toulouse School of Economics (INRA-LERNA) (France), 9 Polytechnic University of Valencia (Spain), 10 International Food Policy Institute (USA)*

**Abstract:** Water management models that lack a user interface, data management, reporting and visualization capabilities are less likely to make an impact on real world water systems. Model platforms can improve this situation; they allow researchers and model developers to link model codes to a generic user interface and data management system. We present an open-source model platform for network-based models called HydroPlatform. The platform links users and their data to water management models through generic import/export functions or custom add-ins. This paper focuses on export functionality, and specifically on a link to the Generalized Algebraic Modeling System (GAMS). HydroPlatform is currently being used for several large-scale modeling projects and add-ins have been built for two generic water management models. Benefits, limitations and future developments of the software are discussed.

**Keywords:** water management models, modeling frameworks, decision support systems (DSS), simulation and optimization models, model platforms, open-source.

## **1. INTRODUCTION**

Although researchers propose and apply many innovative models to help manage water resources, only a few models are regularly used in practice. 'Real-world' modeling studies are data intensive and have time constraints. In addition to the software related to modeling water movement and allocations, real-world water studies require ways to efficiently create, manage and visualize model inputs, scenarios and results. In this paper we call this 'user-software', i.e. software that helps users apply a mathematical model to a real case study. To meet this need, practitioners increasingly use regularly-updated decision support system (DSS) software for water resources studies rather than models developed in research institutions which typically lack user interfaces, data management, reporting and visualization capabilities. State-of-the-art research models are less likely make a significant contribution to water management globally unless significant resources are applied to building user-friendly model interface.

HydroPlatform provides water management model developers a generic user-interface and data management system. Any model with a network structure (node and links) can be linked to the platform regardless of the language or software used to create the model. HydroPlatform allows model developers to quickly and inexpensively link network models to a modular and flexible open-source platform.

To better explain the context for why such a linkage is useful, we briefly describe the structure and objectives of water management models. We then list existing software

options, including model platforms, a term introduced here. We describe the structure and organization of the HydroPlatform software and show how the platform can be used to connect to commercial models and modeling systems such as the Generalized Algebraic Modeling system (GAMS). Finally, limitations, project structure and future developments are briefly discussed.

### **1.1 Water resource management models**

Since Maass et al. (1962), water resource systems have been modeled as networks to represent hydrology, infrastructure (storage, conveyance, treatment, etc.) and diverse water demands. The network structure is straightforward, flexible, parsimonious and works for both simulation and optimization models (Letcher et al. 2007).

Management models use simulation or optimization methods to model the use, flow and storage of water throughout the system under different scenarios, infrastructures or policies. Simulation models predict system performance under a determined set of conditions and rules. Optimization models search through a range of alternatives to find which system solution best satisfies mathematically stated objectives taking into account various physical, environmental, social, political and institutional constraints. The purpose of management models is to support planning and management decisions for system design, expansion and operation.

## **2. SOFTWARE APPROACHES**

There are many software options currently available to run water management models. These options include dedicated water resource decision support systems (DSS), modeling frameworks, commercial modeling systems, and stand-alone software. The boundaries between these approaches sometimes overlap. We describe each type before introducing the model platform approach.

### **2.1 Water resource decision support systems (DSS)**

A DSS is a user-interface built around one or more custom models. The user-interface manages a particular model or modeling process and provides a system to enter, store, and retrieve model data, run the model(s), and view results. Disciplinary water resource DSS have been historically the most popular choice for water resources management studies; some systems have been in use for over 2 decades. Established water resource management DSS include WATHNET (Kuczera 1992), AQUATOOL (Andreu et al. 1996), OASIS (Randall et al. 1997), MISER (Fowler et al. 1999), MODSIM (Labadie and Baldo 2000), RIVERWARE (Zagona et al. 2001), MIKE BASIN (Jha and Das Gupta 2003), CALSIM (Draper et al. 2004), REALM (Perera et al. 2005), WEAP (Yates et al. 2005), WRAP (Wurbs 2005), HEC-ResSim (USACE 2007), WaterWare (Cetinkaya et al. 2008), RIBASIM (WL Delft Hydraulics 2004), AQUATOR (Oxford Scientific Software 2008), and WARGI (Sechi and Sulis 2009).

Advantages of this approach are its conceptual simplicity, the ability to accommodate popular legacy codes and full customization of the user experience around a specific model or set of models. However, this approach forces developers to build, maintain, and extend (as new models are included) the user interface to accommodate existing and new models. Only a handful of DSS exist because software development and maintenance costs are high. As more models or functionality are included to allow integrated assessment, it becomes more effective to use the component-based paradigm (next section).

### **2.2 Modeling and Component-based frameworks**

Modeling frameworks (Argent 2004) or application hosting systems (Rizzoli and Argent 2006) enable building environmental management systems by assembling different components. Such frameworks are designed to integrate models, where modular

components can be assembled to form a custom application. A component-based approach is ideal for building interdisciplinary models as it provides a structured way for different models to interact.

Rizzoli and Argent (2006) and Argent et al. (2006) review modeling frameworks that can be used to build integrated water resource management tools. Examples include the Interactive Component Modeling System (ICMS) (Argent 2004), TIME, the Integrated Modeling Toolkit, TwoLe (Soncini-Sessa et al. 1999) and others.

One potential disadvantage of using comprehensive modeling platforms is they may require that model components be substantially rebuilt to be compatible with the framework. This task has been simplified by the open modeling interface (OpenMI) that allows building wrappers around legacy codes allowing them to be connected to other models (Gregersen et al. 2007). If modelers are willing to adopt a modeling framework and learn the skills required to modify their existing tools for compatibility with a new platform, the rewards in terms of usability and future integration of other models can be large.

### **2.3 Commercial modeling systems**

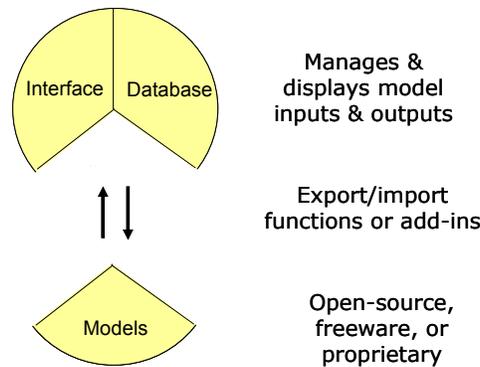
When more model formulation customization is required than what is offered by existing water management DSS, another option is to use generic commercial modeling systems. Both simulation and optimization variants exist. Optimization modeling systems integrate model data, formulation, solution and results definition. These systems separate generically posed mathematical model formulation and the data related to a particular application or “instance” of the model. The ability to flexibly define the data sets for which model equations are applied enable relatively quick development of models. Examples include GAMS, AMPL, LINDO and AIMMS, each of which link model formulations written in high level languages to commercial optimization algorithms called solvers. These systems are flexible, transparent, self-documenting, provide simple links between model formulation and solver solution, and are widely used by researchers and systems analysis practitioners to build custom models. Simulation modeling systems (also referred to as ‘systems dynamics’ software) solve user-defined simulation models with a commercial solver engine. Examples of commercial systems with simulation capability include STELLA, Goldsim, Powersim, AnyLogic, MATLAB Simulink, and others. Some modeling systems allow building custom graphical user interfaces (GUI), while others (e.g. GAMS) do not.

### **2.4 Stand-alone programs**

Traditional stand-alone programs were intended for well conscribed analyses with pre-defined inputs and outputs. Alternatively the term ‘monolithic’ is used by Rizzoli and Argent (2006) to underline the static and less flexible nature of these programs, particularly when it comes to linking them to other programs. Many legacy stand-alone programs are prevalent and effective due to their simplicity and familiarity. Some common examples include MODFLOW, HSPF, SWAT, HEC-RAS, etc.... An effective software solution should be able to accommodate such programs.

### **2.5 Model Platforms**

Models built using commercial modeling systems, stand-alone programs and modeling frameworks often lack easy-to-use interfaces. Model platforms manage, store and display model input and output data. Unlike a modeling system, model platforms do not support model formulation and solution; models are external and built independently (Figure 1). The goal is a software platform to enter, manage and visualize model inputs and results and exchange these with models through export/import functions or add-ins. The term ‘model’ rather than ‘modeling’ platform underscores that model coding does not occur within the platform. Rather the platform hosts existing, external models. The external model no longer requires its own user interface and data management software. If tighter coupling between the user-interface and model is preferred, platform ‘add-ins’ can provide seamless integration (the user need not know that several applications are running).



**Figure 1: Model platform concept: user-software (interface and database) connect to external model(s) via generic functions or add-ins.**

The challenge of efficiently building effective but flexible user interfaces and data management systems for existing water management models is not new. Existing solutions include creating a custom interface from scratch for a specific model or using a modeling system that already has user-interface capabilities. An intermediate solution is using a Geographic Information System (GIS) to store and visualize data that is then passed on to a specific model. These approaches and others are reviewed by Argent (2004). All approaches have advantages and limitations but they all require significant investment by modelers beyond building the model itself. Most water management DSS create a specific user interface to link to a particular model. This technique can be effective but costly. Below we describe HydroPlatform, a model platform software that allows modellers to link a generic user interface to various models thereby decreasing development costs..

### 3 HYDROPLATFORM

HydroPlatform is an open-source, generic model platform for water management network-based models. The model platform manages and displays spatial and temporal model data. Models are run either independently, uncoupled from the platform's user-interface and database, or from within the platform using add-ins.

Project goals :

- Make research models easier to use for real problems
- Help modelers focus on model development, not data management

Software objectives:

- Allow modelers to create, manage and visualize network model data efficiently
- Provide an intuitive graphical user interface (GUI) using object-oriented code
- Maintain generality and independence from any single model or modeling system
- Use open-source components for maintainability, transparency and lower costs

HydroPlatform is an initiative by and for the water management community and its name has been selected for that audience. However, the tool itself is generic and can serve as an interface for any mathematical network model (transport, energy, trade, etc.).

#### 3.1 Capabilities

The completed first phase of HydroPlatform development focused on input, storage, display and export of model data. Node-link networks are built by dragging and dropping node and link objects over a map. The map can be a georeferenced raster or vector data or a non-georeferenced background image for spatial reference. Custom network objects (nodes or links) can be built by defining and adding custom data fields to the object. These fields include parameters, seasonal parameters, time-series, and tables of any dimension.

A database made for one model can be reused for a different model. Imagine an initial database was prepared for Model A but now needs to be used for a similar but different Model B. Model B possesses extra parameters and functions to represent desalination plants. In this case the user would save the Model A database as Model B (using 'Save As'), then go to the Object Type Editor and define a new Object Type called for example 'DesalNode'. Then, new DesalNodes can be added to the existing network, populated with data, and the new dataset can be exported to Model B.

Model data is stored in a database residing either on the analyst's local computer or on a network. This flexibility enables collaborative model building over the internet so that modelers in different locations can work concurrently on a shared system.

Export Functions allow networks defined and data entered in HydroPlatform to be easily exported to external models. Exported data also includes connectivity matrices showing how network nodes in a complex water system are linked. These matrices are automatically generated when the export function is run. External models can be run on any system such as spreadsheets, optimization modeling systems (e.g. GAMS, AMPL, AIMMS, etc.), or custom applications built using programming languages (Fortran, C++, MATLAB, etc.).

Extensions or 'add-ins' allow custom export functions (e.g. model-specific input data export functions) or to 'host' external models as modular add-ins. Add-ins enable tighter coupling between the platform and model if desired. Decision Support System features like special interfaces for managing model use or processing results can be developed as add-ins. This paper does not describe the use of add-ins but focuses on the loose coupling link which uses of export/import functions.

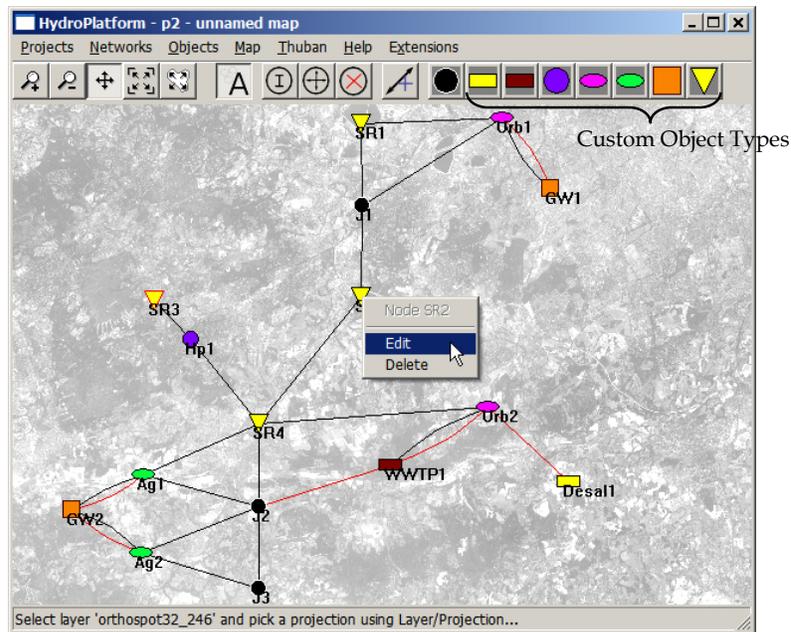
### 3.2 Projects, Networks, Scenarios, Object Types and Objects

HydroPlatform's organizational structure is based on Projects, Networks, Scenarios, Object Types and Objects. Upper-case letters denote words with specific meaning in HydroPlatform. A Project has characteristics such as modeled time horizon, time-step, preferred units, geographic projection, and set of Object Types. A Network is a collection of node Objects connected in a certain way by link Objects (topology). A Project may have more than one Network (e.g. hundreds). Scenarios share a Network but have different data values, i.e. one or more parameters, tables or time-series for one or more node or link Objects are unique. A Network may have any number of Scenarios. When HydroPlatform exports data to an external model, it exports the data content of a particular Scenario.

Each Project has a group of Object Types available for use in that project; each Network is built of individual Objects. In object-oriented programming parlance Object Types are classes whereas Objects are instances of Object Type classes. For example in HydroPlatform 'Reservoir' is an Object Type, whereas 'Reservoir A' and 'Reservoir b' are Objects, i.e. 'instances' of the 'Reservoir' Object Type (class). The instances have the same data structure (data fields) but will differ from each other in their location in the network and possibly the data values in one or more fields. Object Types exist in two main categories ('superclasses'): node (a point) or link (connection between 2 nodes).

Custom definition of Object Types with an easy user-interface is HydroPlatform's most essential contribution to modeling networks. Custom Object Types allows the user to define and HydroPlatform to store and manage data for any network model. Custom Object Types can include any number or grouping of data fields. Data fields can be either parameters, tables, time-series or node references.

Figure 2 shows a simple network implemented in HydroPlatform. The network linking several Objects is drawn over a satellite image. Eight Node Object Types exist in this Project (Figure 2). When new Object Types are deleted or created, they (dis)appear from the Object Type toolbar.



**Figure 2: HydroPlatform screenshot of a simple network including bi-directional links. The toolbar shows this Project has 7 custom Object Types in addition to the default node and link.**

### 3.3 Software implementation

HydroPlatform links an open-source interactive geographic data viewer called Thuban (extensible, multi-platform, multi-lingual) with an open-source database. It is programmed entirely in Python (an object-oriented, multi-platform, open-source scripting language). Data storage is implemented using proven open-source components. It uses SQLAlchemy, the Python SQL toolkit and Object Relational Mapper, and thus supports most database systems such as MySQL, PostgreSQL, SQLite, ...). HydroPlatform uses a set of powerful Python libraries such as Chaco, shapely, networkx, Enthought Traits, etc. for data management to ensure state-of-the-art data manipulation, organization and visualization.

HydroPlatform is free and open-source available under a General Public License (GPL). Add-ins must also be GPL but external models can use any licensing scheme.

### 3.4 Working with a modeling system

Here we describe how HydroPlatform exports model data to GAMS, one of several modeling systems. GAMS consists of a language compiler and numerous open-source and commercial optimization algorithms ('solvers'). The user enters model equations defined over specified indices and data, then GAMS interprets the code and converts the model to matrix-form for solution by a user-selected solver.

A GAMS model consists of 4 'classes'. A class in object oriented parlance is an abstract definition of a group of things with common characteristics. For example, the Object Types described in HydroPlatform (e.g. Desal nodes) could be considered a class. The GAMS classes are: SETS (the indices over which the model is defined, e.g. for a daily model there will be a set  $t$  of all days or a network will have a set of all nodes  $i$ ), SCALARS (fixed constants), PARAMATERS (vector of fixed constants for a set), TABLES (matrices of fixed constants for multiple sets), VARIABLES (decision and state variables, that are unknown and whose values are determined by the solver), and EQUATIONS (mathematical expressions using sets, data and variables).

These classes can have further nested subtypes. For example, the SET object type could include the index  $t$  for time and the index  $i$  for nodes, both of which are themselves sets.

These classes can have further subclasses, e.g.  $desal(i)$  denotes the subset of nodes  $i$  that are desalination nodes. Once subtypes are defined, the instances of subtypes needed for the model can be declared. For example, instances of sub-type  $t$  could be the years 1990 to 2000, or for the  $desal(i)$  subclass, the instances of subtype  $i$  could be desalination plants A, B and C.

In a similar way, the SCALAR, PARAMETER, and TABLE object types can have instances declared for each data item the model requires. For example if an instance of node subclass  $junction(i)$  named 'Junction A' requires an inflow time-series, then Junction A will be associated with a table instance that accepts time-series data.

VARIABLE class instances are variable declarations. Variables are defined using previously defined sets. For example if the model has a storage variable 'Stor' that represents the storage at node  $i$  during time period  $t$ , it could be defined using the  $i$  and  $t$  sets as indices, e.g.  $Stor(i,t)$ .

The object-oriented structure of GAMS easily accommodates water resource network models, for example flow along a network link is defined as  $Q(i,j,t)$ , the flow from node  $i$  to node  $j$  during time-step  $t$ . A network connectivity matrix summarizes the network topology (i.e. which nodes connect to each other) for use in model equations.

HydroPlatform organizes and exports all relevant data in text files or Excel sheets readable by GAMS. The HydroPlatform export function turns all (i) node and link Objects, (ii) the data fields associated with each object, (iii) data for each instance of the objects, and (iv) the network connectivity matrix into forms available for use by GAMS: namely (i) SETS, (ii) PARAMETERS and TABLES indexed by those SETS, (iii) data values for each indexed element in a PARAMETER and TABLE, and (iv) a network connectivity matrix ( $i \times i$ ) whose columns and rows are the SET of all model network nodes. A connectivity matrix value of 1 indicates the associated nodes are linked; 0 indicates other-wise.

In summary, HydroPlatform allows users to build a database customized for the data needs of any network model. The custom HydroPlatform database content is then exported to GAMS using generic export functions where it is read-in, linked to a particular model, and run. The HydroPlatform GAMS export function is a good example of how HydroPlatform works with modeling systems in particular and external programs more generally.

#### 4. BENEFITS AND LIMITATIONS

There are benefits and limitations of the separation of models and interface in model platforms. One benefit of loose coupling is that the modeling and data management software can use different licenses. For example a commercial model (proprietary license) vendor can use HydroPlatform (open-source license) to commercialize a new model or prototype new model formulations. Generally modularity facilitates software project management; model and data manager can evolve separately.

Disadvantages include a potentially less smooth user-experience than with tightly integrated DSS. If export functions are used, HydroPlatform and the external model must be used separately. With add-ins, the model can run from within HydroPlatform, but this setup requires python programming skills. Previously implemented add-ins can serve as templates. For models that are already tightly integrated into an existing proprietary DSS, options for linking to HydroPlatform will be limited unless the model can be isolated.

#### 5. PROJECT STRUCTURE, CURRENT STATUS AND FUTURE

The project is an open-source, international collaboration among numerous researchers and practitioners. We are developing and using HydroPlatform as the single user interface and data management system for our varied and diverse set of existing water management models and models under development.

A working version is available for early adopters. The software download will be distributed through [www.hydroplatform.org](http://www.hydroplatform.org). The project website contains documentation and will eventually work like an app store, showcasing the open-source, freeware, or proprietary water management models that link to HydroPlatform. The project is managed with an online subversion (SVN) repository that stores the wiki, code and bug reports.

The completed phase 1 focused on building network data sets and exporting them to external models using export functions or add-ins. Two generic export functions exist, exporting to text files or to a single spreadsheet; both formats can be read by GAMS. Several GAMS models currently under development have been linked to HydroPlatform including a water supply infrastructure capacity expansion model (South England, UK), a hydro-ecological allocation model for a bird refuge (Utah, USA), a hydro-ecological model investigating hydropower impacts under climate change (California, USA) and a hydro-economic model studying water trading (California, USA). Two add-ins have been completed, generating input files for IRAS-2010, a generalized water resource simulation model (Loucks et al. 1995; Loucks 2002; Matrosov 2009), and the AquaPlan hydro-economic optimization model (Tilmant et al. 2008).

Phase 2 will focus on two tasks: 1) importing model results back into HydroPlatform for storage, analysis and display and 2) upgrading data structure to accommodate stochastic data (i.e. ensembles of inputs and outputs including parameters, tables or time-series, performance measures, or resulting time-series). Phase 2 will also introduce Scenarios; the current proto-type uses only Projects and Networks.

## 6. CONCLUSIONS

A model platform allows users to input, visualize and manage network model input data and results. Users can link their model to a generic user-interface and data management platform and easily apply a model to real systems. This setup contrasts with other prevalent water management software setups such as DSS, modeling systems, or stand-alone programs that typically either provide limited user-interface capabilities or have sophisticated user interfaces but force use of the built-in model.

We have developed a generic model platform called HydroPlatform as an open-source, international collaborative effort. HydroPlatform can be used with any network-based model as formulating and solving models is done external to and independent of HydroPlatform. HydroPlatform uses standard data export functions or customized add-ins to send data entered in the platform to the external model. By helping to input, visualize, and manage network model data, HydroPlatform allows model developers to focus their time and effort on model rather than software development.

## References

- Andreu, J., Capilla, J., and Sanchis, E. (1996). "AQUATOOL, a generalized decision-support system for water-resources planning and operational management." *Journal of Hydrology*, 177(3-4), 269-291.
- Argent, R. M. (2004). "An overview of model integration for environmental application - components, frameworks and semantics." *Environmental Modelling & Software*, 19(3), 219-234.
- Argent, R. M., Voinov, A., Maxwell, T., Cuddy, S. M., Rahman, J. M., Seaton, S., Vertessy, R. A., and Braddock, R. D. (2006). "Comparing modelling frameworks - A workshop approach." *Environmental Modelling & Software*, 21(7), 895-910.
- Cetinkaya, C. P., Fistikoglu, O., Fedra, K., and Harmancioglu, N. B. (2008). "Optimization methods applied for sustainable management of water-scarce basins." *Journal of Hydroinformatics*, 10, 69-95.
- Draper, A. J., Munevar, A., Arora, S. K., Reyes, E., Parker, N. L., Chung, F. I., and Peterson, L. E. (2004). "CalSim: Generalized model for reservoir system analysis." *Journal of Water Resources Planning and Management*, 130(6), 480-489.

- Fowler, M. R., Cook, S. C., and Lumbers, J. P. (1999). "Practical experience in the successful implementation of optimisation systems for water supply management." Computing and Control for the Water Industry, Tynemarch Ltd., Exeter, UK, 1999.
- Gregersen, J. B., Gijbers, P. J. A., and Westen, S. J. P. (2007). "OpenMI: Open modelling interface." *Journal of Hydroinformatics*, 9(3), 175-191.
- Jha, M. K., and Das Gupta, A. (2003). "Application of Mike Basin for water management strategies in a watershed." *Water International*, 28(1), 27-35.
- Kuczera, G. (1992). "Water-Supply Headworks Simulation Using Network Linear-Programming." *Advances in Engineering Software*, 14(1), 55-60.
- Labadie, J. W., and Baldo, M. L. (2000). "MODSIM: Decision Support System for River Basin Management: Documentation and User Manual." Dept. of Civil Engrg., Colo.State Univ., Ft. Collins, Colo.
- Letcher, R. A., Croke, B. F. W., and Jakeman, A. J. (2007). "Integrated assessment modelling for water resource allocation and management: A generalised conceptual framework." *Environmental Modelling & Software*, 22(5), 733-742.
- Loucks, D. P. (2002). "IRAS – Interactive River-Aquifer Simulation for Policy Impact Prediction." *Regional Water System Management: water Conservation, Water Supply and System Integration*, Cabrera, Cobacho, and Lund, eds., A.A. Balkema Pub., Lisse.
- Loucks, D. P., Taylor, M. R., and French, P. N. (1995). "IRAS - Interactive river-aquifer simulation model, program description and operating manual." Cornell University, Ithaca, NY.
- Maass, A., M. Hufschmidt, R. Dorfman, H. Thomas, S. Marglin, and Fair, G. (1962). *Design of Water-Resources Systems*, Harvard University Press, Cambridge, Massachusetts.
- Matrosov, E. (2009). "Interactive River-Aquifer Simulation release 2010 (IRAS-2010) - Application to the Thames Water Resource System," TU München, Munich, Germany.
- Oxford Scientific Software. (2008). "A Guide to Aquator, 1. Application, Version 3.0." Oxford Scientific Software, Oxford, UK.
- Perera, B. J. C., James, B., and Kularathna, M. D. U. (2005). "Computer software tool REALM for sustainable water allocation and management." *Journal of Environmental Management*, 77(4), 291-300.
- Randall, D., Cleland, L., Kuehne, C. S., Link, G. W., and Sheer, D. P. (1997). "Water supply planning simulation model using mixed-integer linear programming "engine"." *Journal of Water Resources Planning and Management-Asce*, 123(2), 116-124.
- Rizzoli, A. E., and Argent, R. M. (2006). "Software and Software Systems: Platforms and Issues for IWRM Problems." *Sustainable management of water resources: an integrated approach*, C. Giupponi, D. Karssenberg, A. Jakeman, and M. P. Hare, eds., Edward Elgar.
- Sechi, G. M., and Sulis, A. (2009). "Water System Management through a Mixed Optimization-Simulation Approach." *Journal of Water Resources Planning and Management-Asce*, 135(3), 160-170.
- Soncini-Sessa, R., Villa, L., Weber, E., and Rizzoli, A. E. (1999). "TwoLe: a software tool for planning and management of water reservoir networks." *Hydrological Sciences Journal-Journal Des Sciences Hydrologiques*, 44(4), 619-631.
- Tilmant, A., Pinte, D., and Goor, Q. (2008). "Assessing marginal water values in multipurpose multireservoir systems via stochastic programming." *Water Resources Research*, 44, W12431, doi:10.1029/2008WR007024.
- USACE. (2007). "HEC-ResSim Reservoir System Simulation User's Manual Version 3.0." USACE.
- WL Delft Hydraulics. (2004). "RIBASIM, Version 6.32." WL Delft Hydraulics, Delft, Holland.
- Wurbs, R. A. (2005). "Modeling river/reservoir system management, water allocation, and supply reliability." *Journal of Hydrology*, 300(1-4), 100-113.
- Yates, D., Sieber, J., Purkey, D., and Huber-Lee, A. (2005). "WEAP21 - A demand-, priority-, and preference-driven water planning model Part 1: Model characteristics." *Water International*, 30(4), 487-500.
- Zagona, E., Fulp, T. J., Shane, R., Magee, T., and Goranflo, M. (2001). "RIVERWARE: A Generalized Tool for Complex Reservoir System Modeling." *Journal of the American Water Resources Association*, 37(4), 913-929.