19th International Congress on Modelling and Simulation, Perth, Australia, 12–16 December 2011
http://mssanz.org.au/modsim2011

Exposing a Hydrological Simulation Model on the Web

B.P. Leighton[a,b], P. Manser[a], D.J. Penton[a,b], J. Shoesmith[a], M.P. Stenson[a], J. Vleeshouwer[a]

[a]CSIRO Water for a Healthy Country National Research Flagship, CSIRO Land and Water, GPO Box 1666, Canberra, ACT 2601, Australia
[b] eWater Cooperative Research Centre, University of Canberra, ACT 2601, Australia
Email: ben.leighton@csiro.au

Abstract: Hydrological simulation models are often implemented as desktop applications. The desktop application paradigm provides some advantages; data is often stored locally, traditional user interface development tools and techniques can be utilised, and deployment is easy in offline environments. However desktop applications cannot make use of a growing number of models and data services provided through web services. The Source Web Server exposes a hydrological simulation model, the eWater Source Integrated Modelling System (Source IMS), as a web service. Delivering modelling functionality as a web service facilitates ready integration with other services and systems. A case study demonstrates the Source Web Server as a means to integrate Source IMS functionality with data processing and visualisation tools.

The rapid development of the World Wide Web has opened up the possibility of leveraging a growing resource of tools and techniques for data processing, user interface development and visualisation. The Source IMS encompasses significant knowledge and modelling capability. This paper discusses enhancing this capability by adding the ability for the Source IMS to interact in new modelling paradigms offered by hydrologic workflows and the Web. Delivering hydrological modelling capability via flexible and simple standards over the Web taps into the existing infrastructure of the Web environment. This infrastructure includes a significant number of useful tools and services for data processing and visualisation.

The Source Web Server is an early implementation of a web service that exposes Source IMS functionality and delivers it into a standardised web environment. The Source Web Server is founded on the Source External Interface. The Source External Interface was developed within the Water Information Research and Development Alliance (WIRADA) in collaboration with the eWater CRC and provides a foundation for alternate interfaces to Source IMS functionality. The Source External Interface has been used to build a command line client and server that provide a practical method of integrating the Source IMS within workflows such as the Australian Water Resources Assessment System (AWRA) under development within WIRADA. The Source Web Server builds upon the command line client and server by extending the use of the Microsoft Windows Communication Framework. It delivers Source IMS capabilities through simple use of the Hypertext Transfer Protocol (HTTP) and provides data in a standard lightweight JavaScript Object Notation (JSON) format.

A HTML user interface to the Source IMS was developed as a case study to examine the integration of Source IMS functionality with web based technologies. The HTML user interface was developed quickly by writing some HTML code that exploiting processing power in existing JavaScript libraries. JSON, jQuery and Google Visualisation Libraries provided much of the required data processing and visualisation functionality. The case study demonstrated that the Source Web Server is a practical and simple means of exposing Source IMS functionality for use in the Web environment. The HTML user interface is a novel, customised, web mashup that is useful for stakeholder engagement. Development of other mashups should be possible using the general purpose services provided by the Source Web Server. The Source Web Server delivers hydrological modelling functionality beyond traditional desktop paradigms. It builds upon the Source External Interface and associated work to deliver Source IMS capabilities into a Web environment providing an abundance of valuable data processing services, tools, and techniques.

Keywords: *River Modelling, web delivery, mashup, JSON, jQuery, WCF, HTML 5*

1. INTRODUCTION

Web technologies provide opportunities for new ways to deliver river modelling functionality. Model developers and water managers traditionally interact with river models through desktop applications. This paper describes the Source Web Server, a new web service for the eWater Source Integrated Modelling System (Source IMS). A case study describes the construction of a HTML user interface to act as a client for the Source Web Server. The case study demonstrates the use of simple and relatively new techniques to integrate the Source Web Server into a Web environment allowing ready leverage of valuable data services and tools.

Many modern systems for simulating water resource planning scenarios in regulated environments are primarily desktop bound applications. Such systems include RiverWare (Zagona et. al. 2001), MIKE Basin[1], MODSIM-DSS[2], Integrated Quantity Quality Model (IQQM) (Simons et. al 1999), Resource Allocation Model (REALM) (Perera 2005, VUT et al. 2000), WathNet (Kuczera 1992) and Source IMS (Welsh et al 2011). At present, none of these packages provides functionality over the web. MIKE BASIN provides inter-program communication; Christensen (2004) reports coupling MIKE BASIN with MIKE SHE using the OpenMI interfaces. In the broader hydrological domain, Huang (2003) describes a web client for a server hosted version of the TOPMODEL rainfall runoff model. Engel et al. (2003) describe a web based interface to L-THIA a rainfall runoff model.

The eWater Source Integrated Modelling System is typically accessed through a Windows based Graphical User Interface. The Source IMS provides a transparent, robust and repeatable approach to underpin water planning and management across Australia (Welsh 2011). The Source IMS aims to form an essential component of Australia's water management strategy (Wallbrink 2008). It incorporates established river and catchment modelling capabilities and integrates additional functionality that will be used to develop the next round of water sharing plans across Australia (Welsh 2011).

A 'Source External Interface' has been recently developed allowing the Source IMS to interact directly with other programs, without using the graphical interface. The Source External Interface provides a general foundation for alternate interfaces to the Source IMS including a way to call the system remotely, across a network. The Source External Interface was required to provide a robust foundation for development of the more specific tool, the Source IMS Command Line Client and Server, a way of interacting with the Source IMS using simple command line instructions.

Penton et al (2011) describe the development of the Source External Interface and associated command line client and server with two case studies. The first case study involves the integration of Source IMS in the Delft Flood Early Warning System (Delft FEWS) workflow for the Bureau of Meteorology Australian Water Resources Assessment Project (AWRA). The second case study involves the integration of Source in Parameter ESTimation (PEST) software for calibration of model parameters (Doherty 1994). Through the Water Act (2007), the Bureau of Meteorology (BoM) is required to regularly produce an Australian Water Resource Assessment (AWRA) report and annual National Water Account (NWA); retrospectively reporting on water availability and water use across the continent. The AWRA system (Van Dijk et al., 2011) is designed to provide retrospective estimation of water balance terms in cases where direct observations are not available, and rely on observations where appropriate (Frost et al., 2011). The command line client and server allow integration of the Source IMS functionality into the DELFT workflow and thus provide river modelling capabilities for the AWRA system.

The Source Web Server builds upon the foundational Source External Interface and the command line client and server to provide access to source IMS functionality through a HTTP interface. The command line client and server utilise the Windows Communication Framework (WCF). The Source Web Server extends the use of WCF responding to a simple set of HTTP requests by executing actions on the Source IMS and returning data in a lightweight standard format (JSON). A case study implements a HTML and JavaScript user interface that demonstrates that, with minimal effort, the Source Web Server allows integration of Source IMS functionality with an established and useful ecosystem of Web based technologies. Coupled with JavaScript, and utilising Google visualisation technologies the case study demonstrates that HTML clients can be used in web mashups (Yu 2008) to provide rapid development of, remote, thin, customized interfaces to existing Source IMS models and functionality.

[1] MIKE Basin is a product of DHI Software http://www.dhigroup.com/.
[2] MODSIM-DSS is a product of Colorado State University http://modsim.engr.colostate.edu/

The case study also demonstrates the potential of web mashup interfaces in the hydrological domain. River models often form a centrepiece in negotiations over changes to water management rules. While river models are rarely accessible to the public, often water managers present the model results for different scenarios to interested stakeholders (e.g. sustainable diversion limits in the Murray Darling Basin Plan). The case study illustrates a tailored web interface potentially useful as a tool for direct engagement with stakeholders.

2. THE SOURCE EXTERNAL INTERFACE

The Source External Interface is a standard set of interfaces for loading information into, saving data out of, running, setting parameters upon and retrieving results from the Source IMS. The Source External Interface was required to provide a robust foundation for development of the more specific tool, the Source IMS command line client and server, a way of interacting with the Source IMS using simple command line instructions. This tool meets the requirements of AWRA, in particular it allows integration of Source IMS capabilities with an automated workflow managed by the Delft FEWS software (Gijsbers 2008) it also provides a tool used in an eWater project integrating the Source IMS with the parameter estimation tool (PEST).

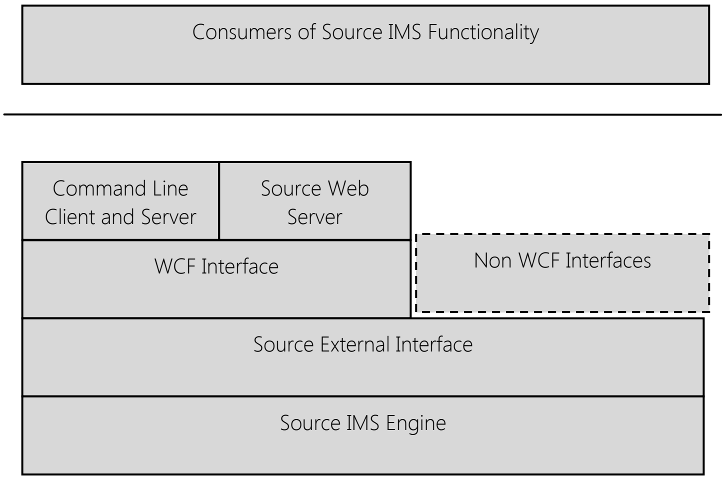

Consumers of Source IMS Functionality
Command Line Client and Server
Source Web Server
WCF Interface
Non WCF Interfaces
Source External Interface
Source IMS Engine

Figure 1. Source External Interface architectural layers

A key component of the command line client and server is the Windows Communication Framework (WCF). WCF provides the ability to expose Microsoft .NET functionality as a service that can be called from elsewhere, where elsewhere can be another part of the same program, another program running on the same machine, or another program running remotely. WCF provides a declarative mark-up mechanism for defining public entry points to software. Using this mark-up, WCF allows developers to implement a single general purpose service based architecture. WCF provides an architectural layer on top of the Source External Interface functionality (see Figure 1.). More refined interfaces can build upon either the WCF component or directly upon the core Source External Interface (see Figure 1). WCF provides general purpose functionality suitable to many but not all implementations of external interfaces. For example, work is progressing to build an OpenMI interface directly up the core Source External Interface. OpenMI compliance will provide the ability to link easily with other OpenMI compliant models (Gergersen 2007).

3. EXTENDING THE USE OF WCF: THE SOURCE WEB SERVER

The Source Web Server builds upon the WCF service contracts used to create the command line client and server. The Source Web Server exposes the WCF service contracts over the web using HTTP. WCF provides simple tools to define mapping between external HTTP requests and internal methods. It also provides a mechanism for specifying a request response format. The Source Web Server uses WebInvoke (Microsoft 2011), to define a response format and URL template for each function provided through the Source External Interface.

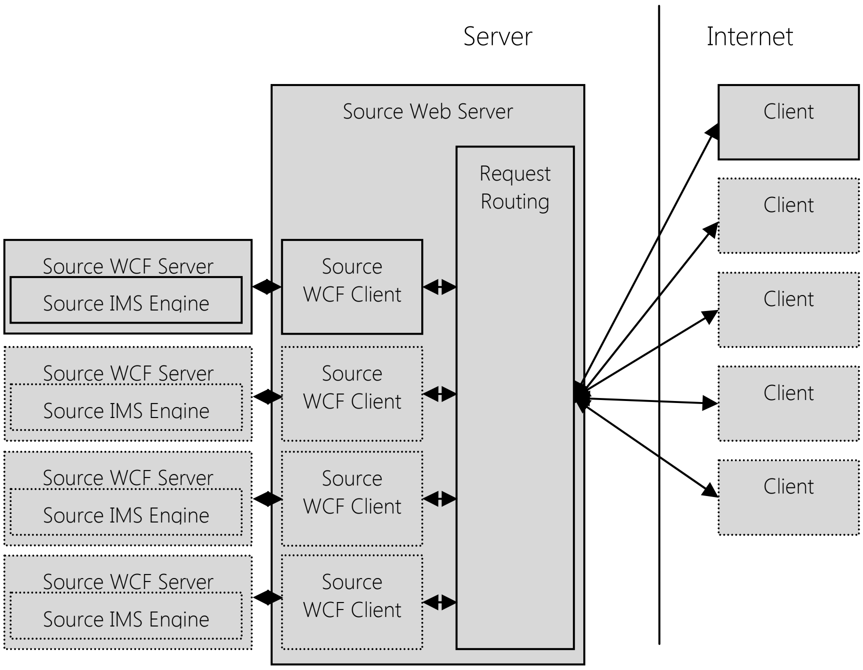


Figure 2. Example of the components of a multi session Source Web Server. n clients can connect to n managed Source servers via the Source

The Source Web Server is accessible to consumers via calls to a standard URL (like a web page), but rather than returning HTML directly it returns data in the JavaScript Object Notation (JSON) format.

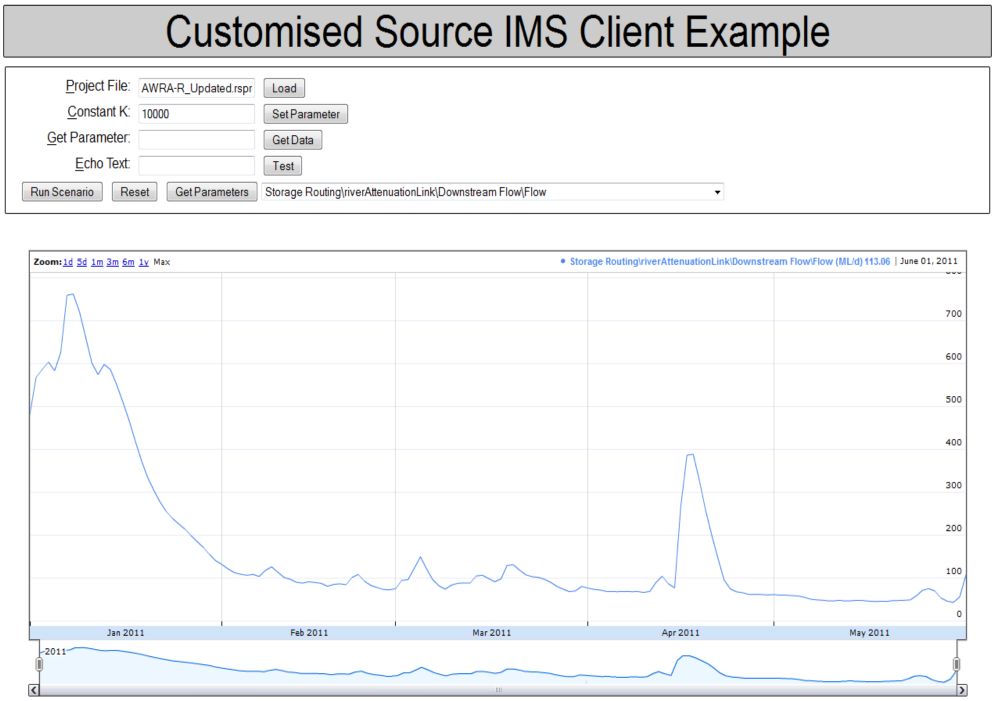
JSON is a text based data interchange format that is lightweight and language independent (Crockford, 2006). .NET through WCF provides in built support for JSON and can readily serialise data in this format. JavaScript (and other) libraries are available for parsing JSON data. The simplicity of JSON and the extensive pre-existing support make it a good choice for rapid development of web applications and integration with existing tools. JSON is similar to XML but more compact in that it wraps data in using more compact text (it does away with the notion of xml tags).

The Source Web Server extends the single session behaviour of the command line client and server by providing sessions to multiple consumers. It presents a single server instance to external users while routing requests to managed Client and Server pairs.

4. A CASE STUDY BUILDING A USER INTERFACE

To test the utility of the Source Web Server and explore a realistic use case, we conceived of a scenario in which a simple HTML user interface could be used for engagement with a stakeholder. In this scenario a simple HTML user interface provides targeted access to a river model.

The HTML user interface to the Source IMS for the case study was created using the jQuery JavaScript library (Bibeault 2008) and an AJAX style approach (Garrett 2005). A JavaScript variable holds a reference to the session identifier and this is used by the client to construct command URLs that are used to call the Source Web Server.


Customised Source IMS Client Example

Project File: AWRA-R_Updated.rspr

Constant K: 10000

Get Parameter:

Echo Text:

Storage Routing\riverAttenuationLink\Downstream Flow\Flow

Figure 3. Customised Source IMS HTML User Interface

This user interface was rapidly constructed using widely available JavaScript and visualisation libraries. The user interface uses standard simple HTML and CSS to build forms and structure and style the layout. JavaScript processes form data into HTTP GET requests that are sent using jQuery commands to the Source Web Server URLs. Result data is returned from the server in JSON format. jQuery dynamically parses returned data and updates components of the page.

Google chart tools[3] provide the HTML user interface with graphs through the Annotated Time Line chart. JavaScript in the HTML further processes JSON data returned by the Source Web Server and configures the Google chart with this data.

While one instance of the HTML user interface is used in the case study, the system supports the ability for multiple clients to interact with the Source Web Server at any one time. Session ids provided as part of each URL call map each instance of the HTML user interface to the appropriate model instance. Any particular instance of a HTML user interface is required to manage a simple session id to identify to the server what river model and state it is connecting to.

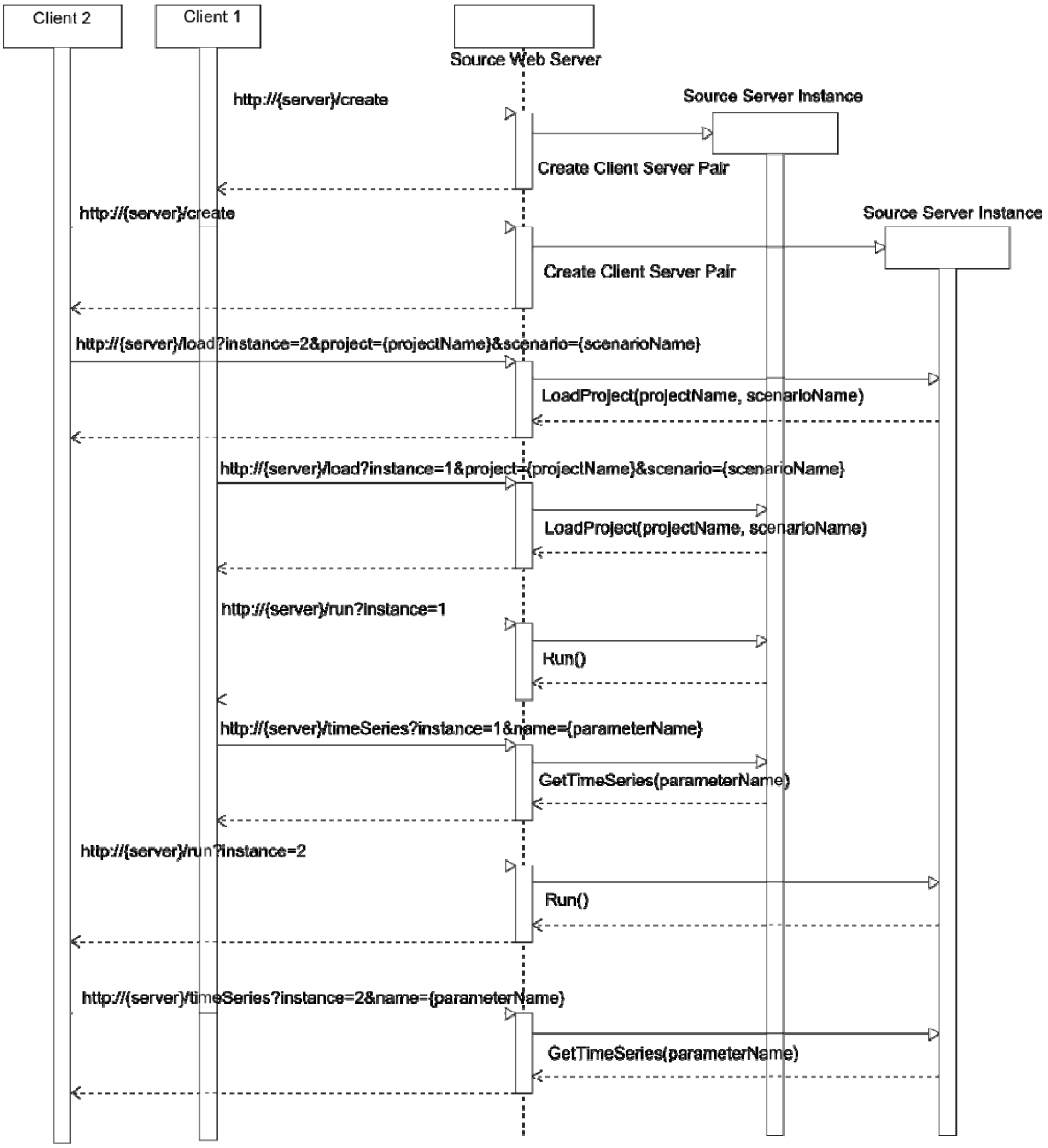

Client 2
Client 1
Source Web Server
Source Server Instance
http://{server}/create
Create Client Server Pair
http://{server}/create
Source Server Instance
Create Client Server Pair
http://{server}/load?instance=2&project={projectName}&scenario={scenarioName}
LoadProject(projectName, scenarioName)
http://{server}/load?instance=1&project={projectName}&scenario={scenarioName}
LoadProject(projectName, scenarioName)
http://{server}/run?instance=1
Run()
http://{server}/timeSeries?instance=1&name={parameterName}
GetTimeSeries(parameterName)
http://{server}/run?instance=2
Run()
http://{server}/timeSeries?instance=2&name={parameterName}
GetTimeSeries(parameterName)

Figure 4 Sequence Diagram for two user interfaces interacting with the Source Web Server

[3] http://code.google.com/apis/chart/interactive/docs/index.html 2011

The HTML user interface takes advantage of the availability of advanced data processing and visualisation technologies. Minimal custom code was needed to develop the client. Less than 100 lines of custom JavaScript functions were required. The HTML code is approximately 70 lines long and the associated CSS code is around 60 lines. The HTML user interface was constructed as a targeted interface to a river model. The HTML user interface provides access to a single river model parameter. It also provides the ability to run the model and visualise the results. A simple river model used in the AWRA project was chosen as an example for study. A routing parameter K was exposed in this model. The HTML user interface provides a means for Stakeholders to interact with model functionality through a target and restricted user interface reducing confusion and focusing on relevant aspects of the river model.

5. DISCUSSION

The early implementation of the Source Web Server uses the WCF infrastructure of the Source External Interface. The Source External Interface provides a well-defined and compact point of extension for the Source IMS. The Source External Interface is compact because it delivers functionality through a small number of exposed functions and it is well defined by a documented C# interface.

A case study was constructed to investigate the utility of the Source Web Server. This case study involved a simple targeted user interface potentially useful for stakeholder engagement. It was found that construction of the user interface was made simple by the leveraging of existing tools within a Web based ecosystem of services and tools that are united by simple standards. The user interface utilised recent visualisation and data manipulation libraries and techniques. The clear demarcation between the Source Web Server and client applications (like the HTML user interface) allows the flexibility paradigm to build interfaces to the Source IMS. HTML user interfaces can be simply, rapidly, and independently constructed.

Novel problems can be addressed through use of the Source Web Server as part of web mashups. Future web mashups could integrate models with data services. For example a client could be created combining river gauge data with modelled results. A mashup could be developed presenting a number of hydrological models embedded as a layer in Google maps. As new services and techniques become available additional unexpected applications should be possible.

The Source Web Server is an early implementation of a system for delivering the SourceIMS functionality through the Web. It is basic, utilising simple extensions to the existing architecture of the Source External Interface and command line client and server tool. The case study illustrates the validity of this approach however more work is needed to validate the long term utility of this system. The robustness and performance of the system have not been extensively tested, particularly its ability to scale with demand. The use of JSON is questionable, it provides a lightweight data format that is simple to use. However it does not deliver rich metadata about the packaged information which limits its ability to integrate in a richer semantic context.

Further work should examine other options for better session management to handle multiple requests. Security is not provided in the implementation outlined; this could be improved by generating more complex session ids, hiding communication inside encrypted https connections, and/or using a different session management system. Currently the Source Web Server only provides access to some of the functionality of Source IMS, for example it excludes the ability to modify the model structure. Furthermore this system directly harnesses separate command line instances of the Source IMS engine.

6. CONCLUSION

The Source Web Server is an early implementation of a system for delivering Source IMS functionality over the web. It provides a basis for further development of a Source IMS web service. The Source Web Server further illustrates the ability to develop new and useful programmatic and user interfaces by building on the framework provided by the Source External Interface and the associated Source IMS command line client and server tool.

A case study describes a custom HTML user interface for the Source Web Server. Development of the user interface involved a minimal amount of custom code. New sophisticated and modern tools for data manipulation and visualisation technologies were utilised in particular the jQuery and Google Visualisation libraries. The user interface readily integrated with these tools indicating a role for the Source Web Server as a way of incorporating the Source IMS with other cutting edge technologies. The user interface indicates, most importantly, that the Source Web Server provides an efficient means to draw upon existing Source IMS capabilities to create a ‘mashup’; a new interface to suit a particular stakeholder group or process.

ACKNOWLEDGMENTS

This work was funded through WIRADA, an alliance of the Bureau of Meteorology and CSIRO's Water for a Healthy Flagship, and the eWater CRC. Additional thanks to Andrew Frost, Joel Rahman and Fareed Mirza.

REFERENCES

Bibeault B., Katz, Y. (2008), jQuery in Action. Book. *Manning Publications* Co. Greenwich, CT, USA

Christensen, F.D , Coupling Between the River Basin Management model (MIKE BASIN) and the 3D Hydrological Model (MIKE SHE) with use of the OpenMI System, 6th International Conference on Hydro informatics (2004).

Crockford D. (2006). The application/json Media Type for JavaScript Object Notation (JSON). Memo, The Internet Society. Retrieved July 6, 2011, from http://www.ietf.org/rfc/rfc4627.txt?number=4627

Doherty, J., 1994. PEST: A unique computer program for model-independent parameter optimisation, in: proc. *Water Down Under '94, Vol. 1*, IE Aust, Adelaide, pp. 551-554.

Engel, B.A., Choi J, Harbor J., Pandey S. (2003). Web-based DSS for hydrologic impact evaluation of small watershed land use changes. *Computers and Electronics in Agriculture*

Frost, A., Lerat, J., Srikanthan, S., Stenson, M., AWRA River System Modelling Requirements and Implementation, *Internal Report*

Garrett JJ. (2005) Ajax: A New Approach to Web Applications. Adaptive Path. Retrieved July 6, 2011 from http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications

Gijsbers, P.J.A., Werner, M.G.F., and Schellekens, J. (2008). Delft FEWS: A proven infrastructure to bring data, sensors and models together: In *Proceedings of the iEMSs Fourth Biennial Meeting: International Congress on Environmental Modelling and Software* (iEMSs 2008). International Environmental Modelling and Software Society, Barcelona, Catalonia, July 2008, pp. 28-36.

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

Huang B. (2003) Web-based dynamic and interactive environmental visualization. *Computers, Environment and Urban Systems, Part Special Issue: Urban Data Management Society*

Kuczera, G. (1992), Water supply headworks simulation using network linear programming, 14(1):55-60.

Liu, Y., Marini, L., Kooper, R., Rodriguez, A., Hill, D., Myers, J., Minsker, B., Virtual Sensors in a Web 2.0 Virtual Watershed. (2008), *eScience, 2008. eScience '08. IEEE Fourth International Conference on.*

Microsoft., (2011) WebInvokeAttribute Class. In *MSDN Library*. Retrieved July 7, 2011, from http://msdn.microsoft.com/en-us/library/system.servicemodel.web.webinvokeattribute.aspx

Penton D.J. Leighton B.P. Stenson M.P. Rahman J.M., Bethune M. Integrating hydrological simulation models into workflow and optimisation software (2011). [Draft Paper]. *MODSIM 11 International Congress on Modelling and Simulation*: Perth, Modelling and Simulation Society of Australia

Perera, B.J.C., B. James, M.D.U. Kularathna (2005), Computer software tool REALM for sustainable water allocation and management, *Journal of Environmental Management*, 77(4):291–300.

Simons, M., G. Podger, R. Cooke (1996), IQQM – A hydrologic modelling tool for water resource and salinity management, *Environmental Software*, 11(1-3):185–192.

Wallbrink, P., (2008). A new national river modelling platform. Water. *Journal of the Australian Water Association*. volume 35, No. 7, November 2008, p 6.

Welsh W.D., Vaze, J., Dutta, D., Rassam, D., Rahman, J., Jolly, I.D., Wallbrink, P. Podger G.M., Bethune, M., Hardy M.J., Teng, J. Lerat, J. (2011). An integrated modelling framework for regulated river systems. [Under Review]. *Environmental Modelling & Software*.

Yu, J., Benatallah, B., Casati, F., Daniel, F. (2008). Understanding Mashup Development. IEEE Internet Computing, 12(5), 44-52.

Van Dijk, A.I.J.M., Bacon, D., Barratt, D., Crosbie, R., Daamen, C., Fitch, P., Frost, A.J., Guerschman, J.P., Henderson, B., King, E.A., McVicar, T.R., Renzullo, L.J., Stenson, M.P. and Viney, N., (2011). Design and development of the Australian Water Resources Assessment system, *WIRADA Symposium*. CSIRO, Melbourne.

Victoria University of Technology and Department of Natural Resources and Environment, Victoria (2000), REALM Worked Examples Manual, http://www.dse.vic.gov.au/vro/water (15/12/2008).

Zagona, E. A., T. J. Fulp, R. Shane, T. Magee, and H. M. Goranflo (2001), RiverWare: A Generalized Tool for Complex Reservoir Systems Modeling, *Journal of the American Water Resources Association*, AWRA 37(4):913-929.