

Providing flexibility in GUI-based river modelling software: Using Expression Editors and plug-ins to create Custom Functions in Source IMS

S.S.H. Kim^{ab}, D. Dutta^{ab}, R. Singh^{ab}, J. Chen^{ab} and W.D. Welsh^{ab}

^a*CSIRO Water for a Healthy Country National Research Flagship, CSIRO Land and Water, GPO Box 1666, Canberra, ACT 2601, Australia*

^b*eWater CRC, University of Canberra Innovation Centre, Canberra ACT 2601, Australia*

Email: shaun.kim@csiro.au

Abstract: River modellers quite often require support to incorporate their custom models into existing tools. The Source Integrated Modelling System (Source IMS) is a modelling and simulation software which allows users to directly integrate complicated expressions into its GUI-based software.

Source IMS is a new generation river system modelling software package for operational and planning purposes in regulated and unregulated river systems developed by the eWater Cooperative Research Centre. One of the key technical user requirements for this software package is a framework that allows users to import customised model functionality into the Source IMS, including the user interface, via the use of plug-ins.

The Source IMS allows the use of programming languages supported by the Microsoft .NET Framework to write algorithms that are executed at each time step. The Expression Editor is a GUI-based tool within the Source IMS that provides the ability to enter an equation to calculate variables such as flow volumes, capacities, losses, gains and water demands for different river system components. The Expression Editor allows users to separately calculate the sub-components of the main expression. This enables the use of circular references, which are commonly used in numerical modelling.

Within the Expression Editor, Custom Functions can be called with the ability to accept multiple variables and parameters as objects to be used during calculation. The variables and parameters can be sourced directly from the user, calculated at each time step via expressions and/or taken from a component of the simulation. Numerical programming techniques are available using the Custom Functions plug-ins. These include the use of multi-dimensional arrays, loops, and mathematical and logical operators and functions. In addition, building the plug-ins using Visual Studio 2010 provides a platform for user-friendly debugging methods. Plug-ins, therefore, represent a powerful mechanism for end users to customise the behaviour of the system and incorporate new scientific knowledge and modelling methods.

Source Rivers, which is a scenario mode of Source IMS designed to support planning and operational aspects of river system management, is used to demonstrate the Custom Functions. The daily Rainfall-Runoff (RR) model, Ge'nie Rural a` 4 param`tres Journalier (GR4J), is imported as a series of Custom Functions in a plug-in written in the programming language, C#. The final expression requires seventeen input variables to calculate inflow at each time step. Eight of these variables are dependent on the previous time step's values in the calculations, requiring circular references in the algorithm. Including GR4J RR model inputs as Custom Functions within Source Rivers makes it suitable for stand-alone applications without dynamic linking with the Source IMS catchment scenarios.

Keywords: *Source Integrated Modelling System (IMS), Expression Editor, Custom Functions, plug-ins, numerical programming*

1. INTRODUCTION

The eWater Cooperative Research Centre has the objective of developing a software tool that has the intrinsic ability to model multiple hydrological scales and environments. The modelling software package is called the Source Integrated Modelling System (Source IMS) and has the ability to dynamically link catchment and river system scenarios. The software package is designed to meet the needs of water management agencies to integrate a wide variety of hydrological modelling methods. The modelling methods available within the software include a variety of rainfall run-off models, groundwater-surface water interaction, hydrologic routing, constituent routing, demand and supply system models, water ownership and resource assessment. The software is designed to provide a common catchment and river modelling platform across Australia. Source IMS has inherited many aspects from existing models. In particular the Integrated Quantity Quality Model (IQQM) (Podger, 2004; Simons *et al.*, 1999; DLWC, 1999), BigMod-MSM (Close, 1986) and Resource Allocation Model (REALM) (Perera, 2005; VU and DSE, 2005).

Existing General User Interface (GUI) based river system modelling packages have been known to allow some flexibility for users to modify operating rules for river networks. REALM allows the editing of maximum capacities for pipes and rivers reaches with the use of many operators and functions, transformation tables, monthly patterns and sub-equations. Capacities can be entered into the “Edit Capacity Relationship” window as an equation that is dependent on a list of variables sourced from other components of the model. The variable can be a particular data type of a reservoir node, a demand node, a carrier or can be taken from a streamflow or demand file. Data types include demand, storage volume, carrier flow, carrier capacity, salinity etc (VU and DSE, 2005).

The IQQM Decision Tree (IDT) is embedded into a variety of node types to allow users to control extractions from a river, releases from reservoirs and maintain flow at any location in a river. The IDT is used to make decisions based on defined conditions at any point in the river system. It is available for inflow, extraction, irrigation extraction, minimum flow, and environmental flow node types. The IDT uses user-specified sets of tabular values that relate to each other known as Flow Control Tables (FCT’s), for example, a relationship between dam releases and storage volume. FCT’s are linked together with arithmetic operators (+, -, ×, or ÷) or a maximum or minimum (Hameed and O’Neill, 2005).

The vision for Source IMS is for it to be widely accepted and used across Australia over at least the next decade. To meet this long-term goal, the software requires some adaptability to account for future changes in technology, water management, policies and scientific modelling methods. Source IMS has therefore been designed as a framework that allows users to import customised model functionality, including the user interface, via the use of plug-ins. The user is able to “plug-in” into many points in the program allowing flexibility for users to provide customised tools and models.

A convenient plug-in point is Source IMS’s Expression Editor (Figure 1). The Expression Editor is a GUI-based tool that provides the capability to define a value using an expression via a range of arithmetic, comparison and logical operators, time variables and functions. This is embedded in a number of model elements (Penton and Gilmore, 2009). Model parameter and input values can be dynamically dependent on the values of other model parameters. Six different component types are available within the Expression Editor (Table 1).

Table 1. Component types available in the Expression Editor.

Component Type	Description
Variables	Local variables defined prior to use in expressions. These can refer to current or any previous time step.
Piecewise linear relations	Used to create a named dataset of X-Y paired values. This is used to perform linear interpolations.
Patterns	Used to create named datasets of repeating time-dependent values, e.g. daily or monthly pattern.
Time series	Data sets obtained from time series files.
Global expressions	Expressions that are visible from all instances of the Expression Editor.
Custom functions	Functions defined externally in a plug-in then called from within the Expression Editor.

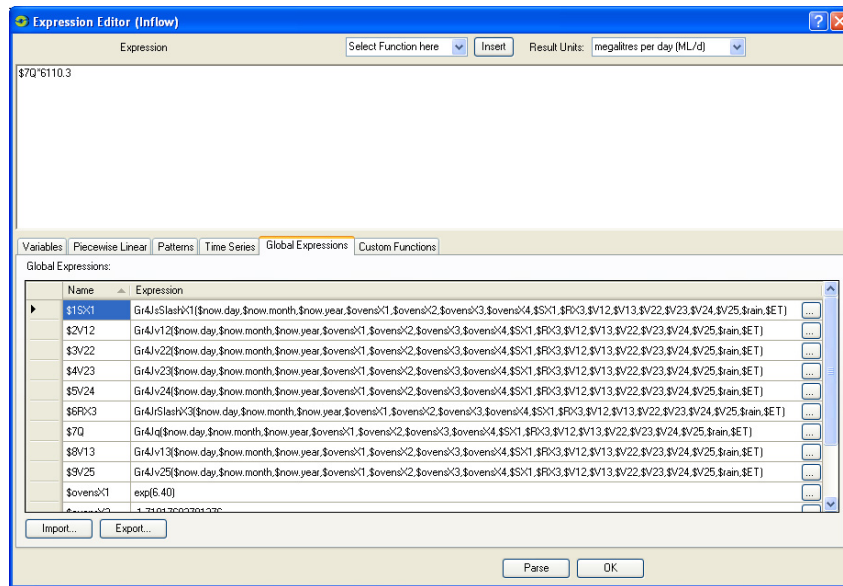


Figure 1. Expression Editor main form for an inflow.

2. CUSTOM FUNCTIONS

One of the Expression Editor’s most powerful features is the Custom Functions component. This allows users to import customised numerical routines to be run at each time step. This functionality is required to allow flexibility for advanced users to incorporate complex algorithms and new modelling techniques. The main benefit of Custom Functions is that they provide additional numerical programming functionality compared to what is available in the Expression Editor. The availability of loops allows for models that use iterative processes such as numerical solvers and optimisers.

Plug-ins must be written in programming languages supported under the .NET Framework 4.0. The .NET Framework is a software framework that runs primarily on Microsoft Windows. It includes a large library and allows language interoperability. Binary files developed by Source must also be referenced for building plug-in projects. Microsoft produces an integrated development environment largely for .NET software called Visual Studio 2010. This is useful for writing, building and debugging the plug-ins. Visual Studio provides many programming tools, such as an advanced code editor and debugging functionality, which allows the user to “step-through” their plug-in code line-by-line as it is running.

Once built, plug-ins are loaded into Source IMS and the functions will become available in the Expression Editor. Custom Functions can be called by entering the function name and the input parameters. Each Custom Function can take several input parameters, including single dimension arrays, but can only return scalar double precision numbers.

3. CASE STUDY: OVENS BASIN

3.1 Site Overview

The Ovens basin, which is in north-eastern Victoria, covers 7,813 km². The basin is based around the Ovens and King rivers. It is bounded to the east and north by the Murray River and to the west by the Goulburn-Broken region and forms the southern edge of the Murray-Darling Basin (MDB). The region’s surface water system terminates at the confluence of the Ovens and Murray rivers. The Ovens River is one of the last largely unregulated rivers in the MDB with high flows close to the natural magnitude and frequency. Significant proportions of the floodplain are inundated in winter and spring in most years.

About one-third of the Ovens region is located within the highlands of the Victorian Alps and includes terrain with steep relief, the valleys of the Ovens River system and open plains. Mean annual rainfall is 1004 mm varying from nearly 1600 mm in the south to 550 mm in the north. Rainfall varies considerably between years, but winter is typically the wettest season. Despite this variability, the region’s average annual rainfall has remained relatively consistent over the past 112 years, except that the mean annual rainfall over the past ten years is about 11 percent lower than the long-term mean.

3.2 GR4J Rainfall-Runoff Modelling using Custom Functions

Source Rivers is a scenario mode of Source IMS designed to support planning and operational aspects of river system management. It is used in the case study to demonstrate the Custom Functions. The daily Rainfall-Runoff (RR) model, Ge'nie Rural a` 4 parametres Journalier (GR4J) (Perrin *et al.*, 2003), is imported as a series of Custom Functions in a plug-in written in the programming language, C#. The Expression Editor performs the series of calculations at each time step. The final expression requires seventeen input variables to calculate inflow at each time step.

Four component types of the Expression Editor were used in the RR model: Global Expressions, Variables, Time Series and Custom Functions. The calculations of GR4J are contained within the Custom Functions. Four input parameters, maximum capacity of the production store (x_1), groundwater exchange coefficient (x_2), maximum capacity of the routing store (x_3) and base lag of the unit hydrograph (x_4), were used for each Custom Function. A series of IF Statements were used to determine the ordinates of the unit hydrograph from x_4 (Figure 2). The rest of Perrin *et al.* (2003)'s method was implemented using basic arithmetic with conditions to use initial values if on the first time step.

The final solution is composed of a series of Global Expressions. Four of the Global Expressions are independently calculated. The rest of the Global Expressions call a Custom Function at each time step using input from rainfall and potential evapotranspiration Time Series, local Variables and the independently calculated expressions. Importantly, the Expression Editor is designed to determine the dependencies of the listed Global Expressions so that they are calculated in the correct sequence starting from the independently calculated expression.

The previous time step results of eight of the Global Expressions are required in order to calculate the new parameters for each time step. Circular references are implemented by using the local Variables to refer to the previous time step's Global Variable results. The local Variables are then the input parameters for the next Global Expression calculation. Figure 3 demonstrates the flow of data between components and between time steps within the Expression Editor.

A warm up period of one year was used at the beginning of the simulation for the RR model. Figure 4 displays the time series for both modelled and observed flow for the Ovens Basin. The time series are reasonably well matched with peaks and troughs lining up and similar magnitude of flow volumes for high and low flow periods. A strong linear correlation is displayed in a scatter plot of the GR4J RR model results and the observed flow data (Figure 5). The Pearson product-moment correlation coefficient and Nash-Sutcliffe model efficiency coefficient was calculated to be 0.95 and 0.88, respectively. It can be seen that running the GR4J RR model with calibrated parameters is a suitable method for modelling of the Ovens Basin.

```
...
double s11;
if( 1.0 < x4 )
    s11 = Math.Pow( 1.0 / x4, 2.5 );
else
    s11 = 1.0;

double s12;
if( 2.0 < x4 )
    s12 = Math.Pow( 2.0 / x4, 2.5 );
else
    s12 = 1.0;

double s13;
if( 3.0 < x4 )
    s13 = Math.Pow( 3.0 / x4, 2.5 );
else
    s13 = 1.0;
...
double hu11 = s11;
double hu12 = s12 - s11;
double hu13 = s13 - s12;
...
```

Figure 2. Calculation of the ordinates of the unit hydrograph for the GR4J RR model.

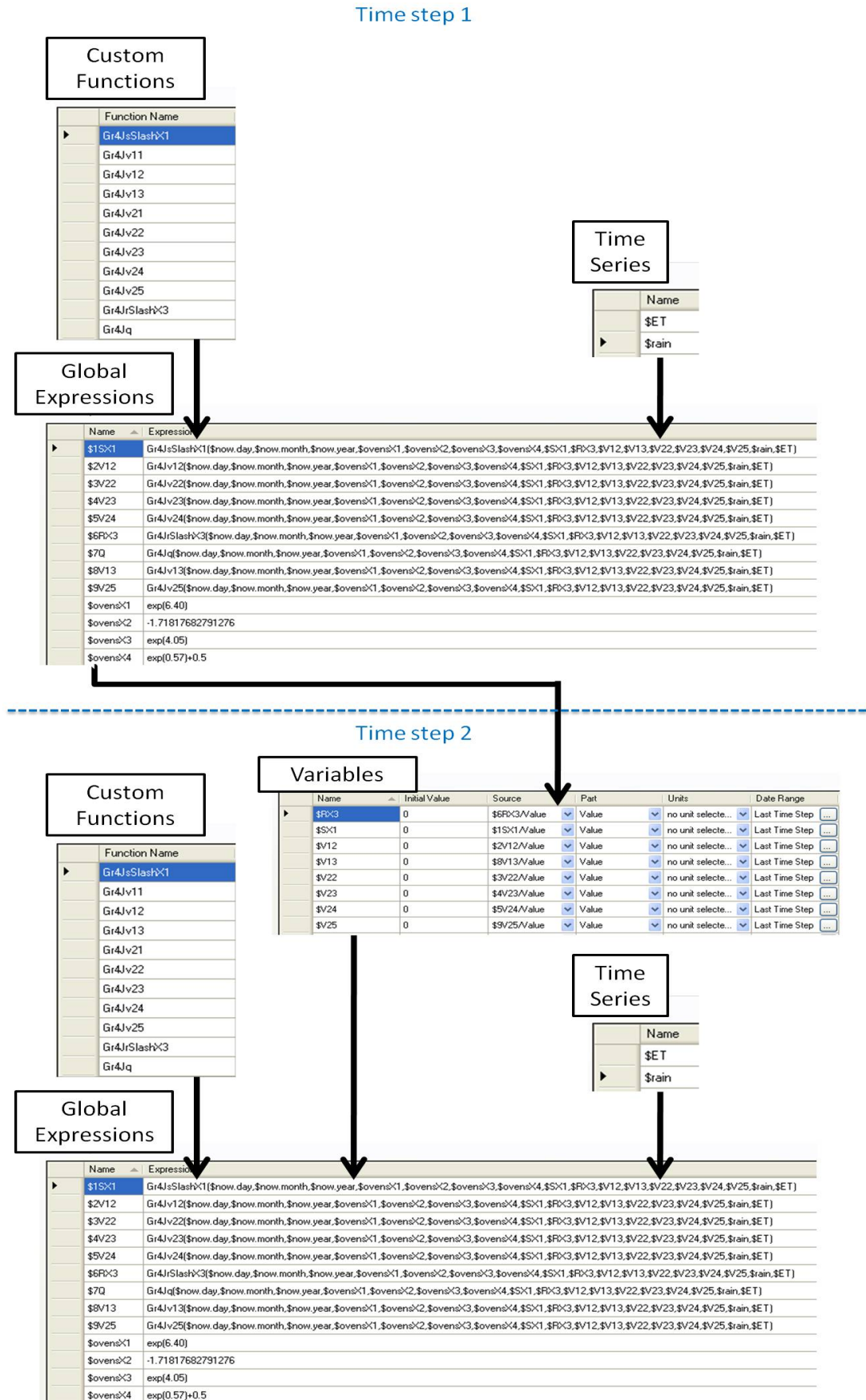


Figure 3. Calculation of expressions during the first and second time step using Expression Editor components Custom Functions, Time Series, Global Expressions and local Variables.

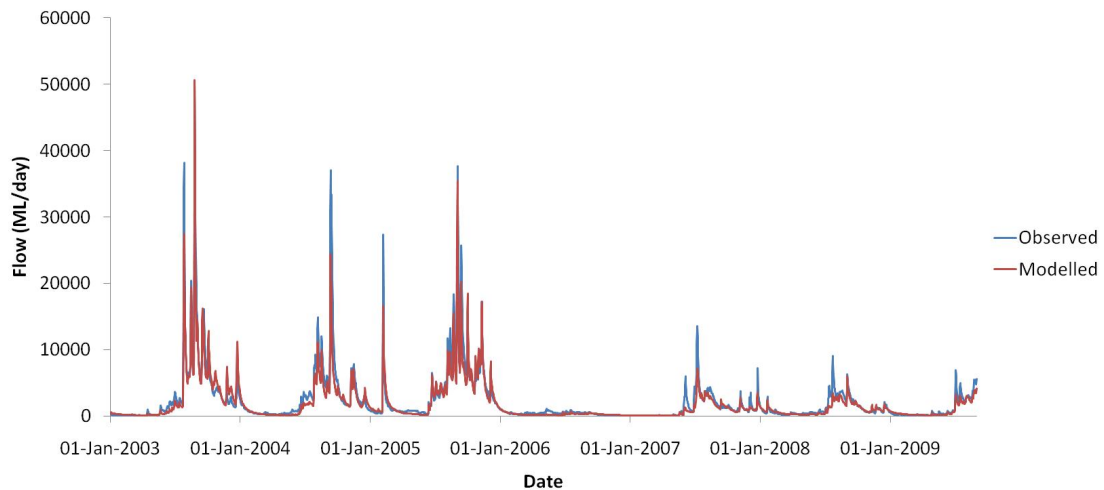


Figure 4. Time series of GR4J RR model results compared to observed flow.

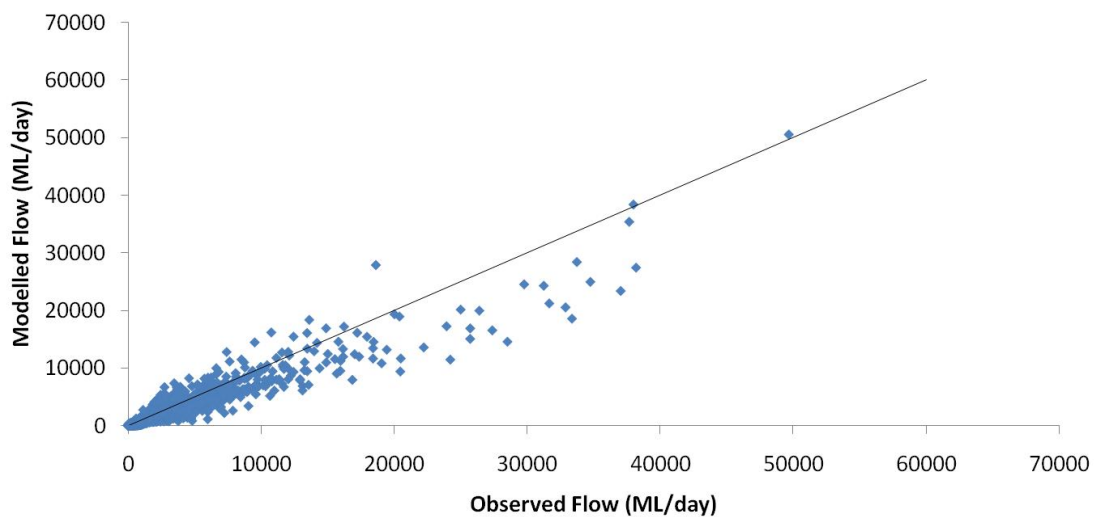


Figure 5. Scatter plot of GR4J RR model results compared to observed flow.

4. DISCUSSION

The GR4J RR model imported via the Expression Editor Custom Functions in this case study used pre-calibrated parameters. The same GR4J RR model used in this case study could have been incorporated using a variety of different methods, including using Global Expressions to calculate the algorithm or dynamically linking the inflow node in a Source IMS Rivers scenario with a Source IMS Catchments scenario. In general, such flexibility is not available from GUI-based software packages especially to users without an information technology or computer science background.

There are advantages in being able to add additional component models to Source IMS as numerical programs. The task of converting complicated numerical routines into expressions can be reduced by using Custom Functions, and this may save further time with the use of Visual Studio 2010's debugging capabilities. The .NET Framework and The Invisible Modelling Environment (TIME) (Rahman *et al.*, 2003) can provide comprehensive libraries useful for hydrologists who require functions that aren't already in the Expression Editor, such as error functions ($\text{erf}(x)$, $\text{erfc}(x)$) for diffusive groundwater models. In addition, the ability to use loops opens up opportunities to incorporate complex numerical solvers and optimisers through iterative processes.

The corresponding functionality in other popular river system modelling packages, such as REALM and IQQM, does not allow the same level of flexibility as Source IMS's Expression Editor. For example, users of REALM are able to define the capacity of carriers, which allows usage of multiple network sources, sub-equations and useful functions and operators. However, this functionality is not

Kim *et al.*, Providing flexibility in GUI-based river modelling software: Using Expression Editors and plug-ins to create Custom Functions in Source IMS

available for creating inflows and therefore RR models would have to be modelled external to REALM and then input through a time series file. IQQM has similar functionality in the IDT that is embedded in multiple node types. Again, the use of the Expression Editor in Source IMS is more generic and is embedded in many more places. Unique features of the Expression Editor that maximise usability are the ability to create and source Global Expressions, and to use local Variables that refer to user-specified time steps. Source IMS is also different from both REALM and IQQM, since neither provide a simple framework to import customised models through the use of plug-ins.

5. CONCLUSION

The Expression Editor is embedded in many Source IMS elements and provides more flexibility and usability than REALM's carrier capacity editor and IQQM's IDT's. Custom Functions allow the use of numerical programming methods not normally available in GUI-based river system models. The ability to include Custom Functions in the Source IMS allows users to build additional functionality, such as Rainfall-Runoff models, into their river and catchment models.

ACKNOWLEDGEMENTS

We thank Dave Penton (CSIRO) for all his contributions to the development of the Expression Editor.

REFERENCES

- Close, A.F. (1986), Computer Modelling of the River Murray, In Hydrology and Water Resource Symposium, Griffith University, Brisbane, November 1986.
- Department of Land and Water Conservation (1999), Integrated quantity-quality model, Reference Manual, Australia.
- Hameed, T. and R. O'Neill, (2005), River Management Decision Modelling in IQQM, MODSIM 2005 International Congress on Modelling and Simulation, December 2005, pp. 1957-1962.
- Penton, D.J. and Gilmore, R. (2009) Comparing software for modelling the management rules that river operators implement. MODSIM 2009 International Congress on Modelling and Simulation, July 2009, pp. 3865-3871.
- 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.
- Perrin, C., Michel, C. and V. Andreassian, (2003), Improvement of a parsimonious model for streamflow simulations. *Journal of Hydrology*, 279, 275-289.
- Podger, G.D. (2004), IQQM Reference manual, Department of Infrastructure, Planning and Natural Resources.
- Rahman, J.M., Seaton, S.P., Perraud, J-M., Hotham, H., Verrelli, D.I. and Coelman, J.R. (2003), It's TIME for a new environmental modelling framework, Proceedings of MODSIM 2003, (4), 1727-1732
- 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.
- Victoria University and Department of Sustainability and Environment, Victoria (2005), REALM User Manual, <http://www.water.vic.gov.au> (31/3/2009).