CATCHMENT MODELLING TOOLKIT

FRAMEWORKS, MODELS AND MODULES

The Cooperative Research Centre for Catchment Hydrology is developing a new generation of catchment models and modelling support tools, integrated within a system of software known as the Catchment Modelling Toolkit. To promote awareness of the Toolkit, we are releasing a series of Information Bulletins. This Bulletin defines some popular terms and design concepts related to the Toolkit.

BULLETIN NO.2
FEBRUARY 2002

A MAJOR PROJECT IS UNDERWAY TO BUILD A TOOLKIT THAT WILL IMPROVE THE STANDARD AND EFFICIENCY OF CATCHMENT MODELLING

MODELLERS HAVE ASKED FOR BETTER TOOLS

Relative to the recent advances in computing hardware and software, the sophistication of catchment modelling software has barely changed in the last decade. One of the reasons for this is that most catchment model development has been carried out by environmental scientists and engineers with minimal computer science training. The models they develop rarely make use of the power and efficiency of modern software engineering methods. Because of this, most catchment models are difficult to understand, use and adapt, leading to frustration amongst model users and those wishing to build new models. Not surprisingly, most model developers and users desire a better suite of tools.

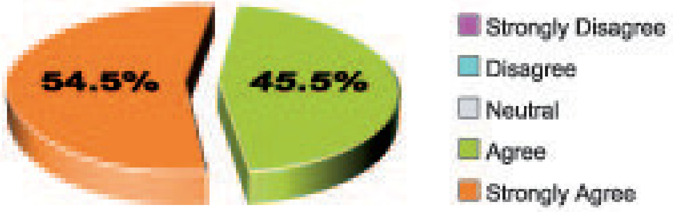


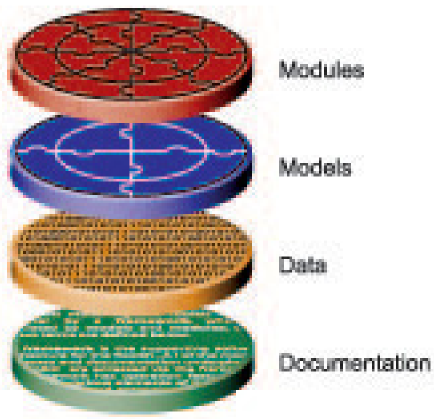
Figure 1: *Responses from a typical cross-section of model developers to the statement "I would like to have access to a model development environment that makes the task of model coding easier".*

FRAMEWORKS, MODELS AND MODULES

The Toolkit is a system of software intended to improve the standard and efficiency of catchment modelling. The Toolkit is populated by models and modules, with most of the modules being held together by a framework. The Toolkit also contains tools for the documentation and management of models and modules. These various terms are defined below.

A **model** is a coded representation of a biophysical or socioeconomic system, which uses input data to calculate a defined set of output results. An example is a flood-forecasting model; this takes rainfall and soil moisture data as inputs and produces an output of water levels at various points within a stream network.

A **module** is a specific component of software that a model uses. An example is a flow routing module within a flood-forecasting model; this transfers parcels of water from one stream segment to another, allowing for time delays and transmission losses. Another example is a time-series graphing component; this might be used to display rainfall or runoff data in several different models.



A **framework** is the supporting software architecture for much of the Toolkit. The Toolkit will contain some stand-alone components – applications that do a particular task well, and which are not commonly linked with other components to form a model. Much of the remaining Toolkit activity, such as building and linking modules to form models, is done using the framework.

Figure 2: *Elements of the Catchment Modelling Toolkit.*

SELECTING A FRAMEWORK

In building the Toolikit, our first challenge has been to find a framework that could be used to build and support modules and models. Some of the main features we sought in a framework were:

- Availability on the PC platform
- Support for module-based model development
- Incorporation of modern software design principles
- An ability to host models written in different programming languages
- An ability to host models of moderate computational complexity
- Advanced data visualization capabilities
- Integrated data and documentation support
- Free access to source code

We are not alone in this search as several other modelling groups around the world (including some hydrology groups) are building frameworks to improve the utility and ease of development of environmental models. Some promising examples of frameworks that we evaluated are:

SME – Spatial Modelling Environment (University of Maryland).
A UNIX-based system, primarily used for ecosystem modelling to date.
http://iee.umces.edu/SME3/

DIAS – Dynamic Information Architecture System (Argonne National Laboratory, Illinois).
A UNIX-based system, used for the Object-Oriented Integrated Dynamic Landscape Analysis and Modelling System, and a range of models outside of the catchment arena.
http://www.dis.anl.gov/DIAS/

ArcGIS and the ArcGIS Hydrology Data Model (ESRI, California).
Modelling software that links to the ARC series of GIS products. http://www.esri.com

GFW – General Framework Water (Institute for Inland Water Management and Waste Water Treatment, The Netherlands).
A hydrologic modelling system in early development that shares many concepts and practices with the Toolkit project. http://waterland.net/riza/aquest/index.html

We evaluated two other frameworks, both developed in Australia by parties to the CRC for Catchment Hydrology. These include:

ICMS – Interactive Component Modelling System (CSIRO Land and Water)
Based on Borland Delphi, ICMS has been used for a variety of small hydrological modelling applications. A great strength of ICMS is that it allows developers with limited programming experience to build models by linking components from a library.
http://www.cbr.clw.csiro.au/icms/

Tarsier (California State University and CSIRO Land and Water).
Based on Borland C++ Builder, Tarsier has been used for a variety of more complex modelling problems. Strengths of Tarsier include its speed and advanced visualization capabilities.
http://earthsystems.monterey.edu/~tarsier

None of the six frameworks we evaluated satisfied all of our requirements, but ICMS and Tarsier met most of them. We have thus selected ICMS and Tarsier for use in the Toolkit, and will further develop both frameworks during 2002 so that they better meet our needs. These two frameworks will be described further in the next Toolkit Bulletin.

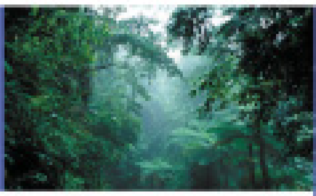
THE YEAR AHEAD

We are now in the final year of the first three-year phase of the Toolkit project. During 2002 we will focus our efforts on:

- using our frameworks to co-develop modules and models with researchers
- training model developers in the use of Tarsier and ICMS and modern software engineering practices
- developing systems for the selection and management of modules and models within the Toolkit
- developing standards for documentation of modules, models and model parameter sets within the Toolkit

As new Toolkit resources are developed, they will be posted on the Toolkit web site at www.catchment.crc.org.au/toolkit. During 2002 we will also plan the second three-year phase of the Toolkit project to commence in January 2003. This second phase will focus on populating the Toolkit with a comprehensive set of modules and models to make holistic catchment prediction a reality.

ICMS AND TARSIER...
TWO MODELLING FRAMEWORKS TO MAKE MODELS MORE POWERFUL AND EASIER TO DEVELOP



RE-USE OF MODEL CODE WILL REDUCE THE TIME AND COST OF MODEL DEVELOPMENT

CATCHMENT MODELS, NOW AND IN THE FUTURE

How will the Toolkit make a difference to the way that catchment modelling is conducted? Catchment models have traditionally been built as islands. They are usually designed to perform a very specific task, more often than not in an idiosyncratic fashion. They generally use unique input and output formats, and rarely link to standard databases. In most cases, the models cannot be linked to other models without significant code modifications. One of the main aims of the Toolkit project is to break this mould, by promoting a module-based approach to model development.

In the past, a model was an immutable block of code developed for one purpose and applied to many others. The model was a solution seeking a problem. In the future, a model will be the result of a process where appropriate modules are selected and combined to fit the modelling solution to the problem.

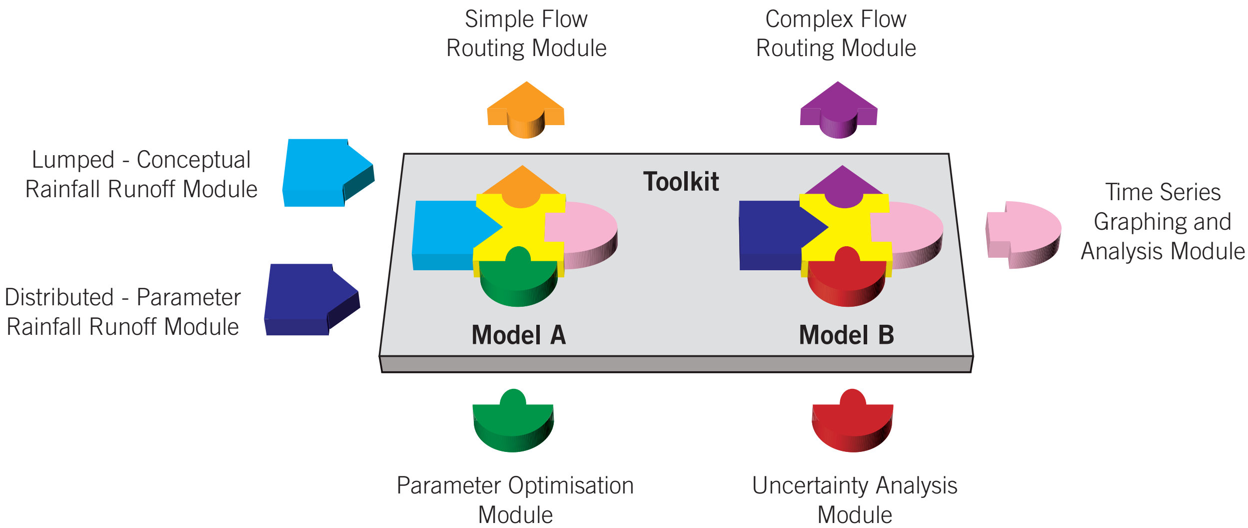
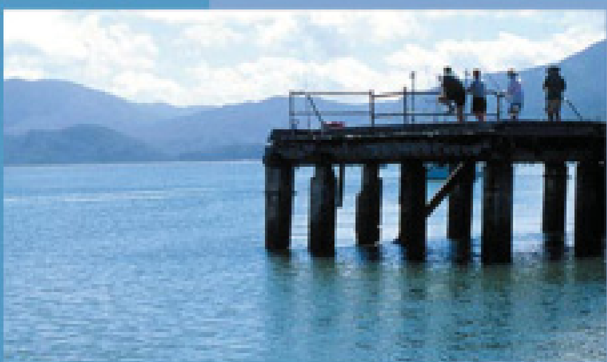


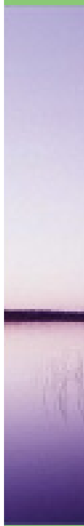
Figure 3: The relationship between models and modules in the Toolkit, showing how two different catchment models (A and B) could be built from a module library. The modules are selected according to the scale and complexity of the modelling problem.

Module-based programming is now regarded as 'best practice' in software engineering circles, and is being promoted through the use of object-oriented programming languages such as Java, C++, Delphi and Visual Basic. In the context of the toolkit, a module-based approach should lead to significant cost savings to the catchment management industry and research community by promoting re-use of code and reducing duplication. Another advantage is that modelling support modules such as data visualization tools, optimization routines and statistical analysis tools will be common across different models. This should reduce the training burden and encourage more responsible use of models.



THE TOOLKIT PROJECT TEAM

The team is led by Robert Argent from Melbourne University and includes Rob Vertessy, Susan Cuddy, Joel Rahman, Shane Seaton, Frances Marston, Fred Watson, Daniel Figucio and John Coleman (CSIRO Land and Water), Roger Braddock (Griffith University), John Ruffini (NRM, QLD), Alex Lau and Christian Maul (NRE, VIC), Mark Littleboy and Geoff Podger (DLWC, NSW) and Roger Hadgraft from Monash University.



GETTING INVOLVED

You can register your interest on-line at www.catchment.crc.org.au/toolkit. You will then be sent regular updates on the progress of the Toolkit project.

Alternatively, you can contact:

Dr Robert Argent
Cooperative Research Centre for Catchment Hydrology
Department of Civil and Environmental Engineering,
The University of Melbourne, Victoria 3010
t 03 8344 7115 **f** 03 8344 6215
R.Argent@civag.unimelb.edu.au

Dr Rob Vertessy
Cooperative Research Centre for Catchment Hydrology
CSIRO Land and Water
GPO Box 1666, Canberra ACT 2601
t 02 6246 5790 **f** 02 6246 5845
rob.vertessy@csiro.au

COOPERATIVE RESEARCH CENTRE FOR
CATCHMENT HYDROLOGY

www.catchment.crc.org.au/toolkit