20th International Congress on Modelling and Simulation, Adelaide, Australia, 1–6 December 2013
www.mssanz.org.au/modsim2013

Comparative code verification using redundancy in a system for national scale hydrological modelling

B. Leighton [a], **D. Penton** [a], **M. Stenson** [a], **P. Manser** [c], **J-M. Perraud** [a], **J. Vleeshouwer** [a], **D. Collins** [b], **R. Bridgart** [a], **F. Mirza** [a], **and S. Kim** [a]

[a] *CSIRO Land and Water,* [b] *CSIRO Information Management and Technology,* [c] *Manser Services*
Email: Ben.Leighton@csiro.au

Abstract: The Australian Water Resources Assessment (AWRA) system (Stenson et al. 2011, van Dijk et al. 2011) was developed to support the production of national scale water resources assessments and accounts by the Australian Bureau of Meteorology. This paper examines a comparative verification process for identifying errors in the development of the river modelling component of the AWRA system.

The AWRA system is comprised of integrated modelling components representing the major parts of the terrestrial water balance, including landscape, river and groundwater processes. Several versions of each of these modelling components have been developed to support different use cases including: testing model improvements, optimizing model parameter sets, simple and stable deployment, high usability, and data model fusion. To meet these use cases there is some duplication in behavior across AWRA versions.

AWRA river modelling consists of separate calibration and operational systems. Across these systems equivalent data handling, execution sequencing, and model connectivity concepts are duplicated. Duplication can increase: the overhead maintaining synchronization, the logistics required to track duplicates, and the number of software bugs and conceptual errors introduced during development (Li 2006).

The duplication and complexity in AWRA increase the risk of bugs in code and in data pre-processing and management. Ensuring AWRA is numerically correct was an important part of development efforts. We describe a software verification technique that leverages the duplication present in AWRA to improve quality. For the purposes of this paper, verification of hydrological models is a set of processes that identify errors in software implementations and ensure the implementation matches the specification. Verification can reduce errors in software and thus can improve quality. Verification differs from model validation which assesses the performance of the model for its intended use. (Sargent 2005). A high level verification approach is function testing that checks the program against external specifications (Myers 2004). Freebairn et. al. (2005) describe a general purpose system that employs a variation of this technique for assessing the quality of a ported hydrological model. A similar strategy was used to create a comparative verification process to improve the quality of the AWRA river modelling systems.

Fundamental in the comparative verification process were comparative tests. These tests helped identify software bugs and issues related to the management of parameters and inputs in the operational and calibration implementations and were automated to provide continuous quality monitoring. The comparative tests compared modelled results checking that, for equivalent inputs, results matched across implementations. We found that, during the development of a modelling system, where there were duplicate components, errors in software structure, implementation, and specification could be identified and resolved through a comparative verification process.

***Keywords**:* *Software testing, Hydrological modelling, Water resource assessments, Redundancy*

1. INTRODUCTION

The Australian Water Resource Assessment (AWRA) system was developed as part of the Water Information Research and Development Alliance between Australia's Commonwealth Scientific and Industrial Research Organisation (CSIRO) and the Australian Bureau of Meteorology (BoM). AWRA generates and combines modelled information with observations to produce accounts and assessments of Australia's water resources across the continent. AWRA models encompass landscape, river, and groundwater processes. Outputs of these models are used to inform assessments of water at varying spatial and temporal scales across the Australian continent. The AWRA system is an umbrella term for an integrated heterogeneous set of sub models and modelling systems. These conceptual models of landscape, river and groundwater processes are expressed in code.

To meet different use cases there are multiple functionally similar or equivalent implementations of some AWRA components. For example operational and calibration systems have differing use cases: AWRA operational systems provide modelled outputs used for the production of national water accounts and run in an operational context providing speed, stability, extensibility, state persistence, the ability to be resumed, and high usability. AWRA calibration systems include separate river and landscape calibration systems designed to allow rapid testing of modelling approaches and exploration of model parameters. The calibration systems are optimised for speed but cannot seamlessly resume after interruption. Maintaining consistency between the duplicated aspects of the two systems was difficult and increased risk.

The operational AWRA system runs as a workflow orchestrated through Delft-FEWS, a workflow engine specialised for flood modelling but useful for a variety of operational hydrological workflows (Gijsbers 2008). This system integrates a gridded national landscape and groundwater model with a composite model consisting of many lumped river reach models. Parallel but separate calibration systems produce parameters used in the operational AWRA Systems. AWRA River modelling (AWRA-R) is part of the broader AWRA modelling system and is responsible for the modelling of water in river reaches, storages and on the floodplain. This paper will describe parts of AWRA-R that perform the same functions in the operational and calibration systems and show how a comparative verification process utilised this duplication to help mitigate risk.

2. AWRA RIVER MODELLING IN CALIBRATION AND OPERATIONAL SYSTEMS

AWRA-R is a composite model consisting of many river reach models. The river reach models run independently when sufficient observed input data are available. However, when there is insufficient input data, upstream river models provide inputs to downstream river models. Thus AWRA-R river reach models are connected to each other and executed in hydrological sequence. Development of AWRA version 3.0 consolidated the river modelling component so that a single copy of the core river model code was used between the calibration and operational systems. Prior to this in version 2 the operational core river reach model utilised the eWater Source Integrated Modelling System (Penton et al. 2011). In version 3 both the operational and calibration systems share a core C river reach model. Factors including development schedules, team boundaries, language preferences and system requirements meant that the calibration and operational systems were developed separately thus use different code to wrap the shared core C model code. The calibration system wraps this core river model in R code (Lerat 2011). The operational system wraps the core river model in C# code. While the core river reach model is reused between the operational and calibration systems the other infrastructure required in the broader composite river model is duplicated. Duplicate behaviour includes hydrological sequencing, data pre and post processing, input, and output.

3. SOFTWARE DUPLICATION AND REDUNDANCY

Software often contains redundant or duplicated code (Kasper 2006). There are different kinds of redundancy and duplication: Code clones are similar or identical sections of code that are typically produced by copying and pasting existing code (Koschke 2007); Redundant code is code that is semantically equivalent. Redundancies can occur accidently or through design and may consist of copied code or share little or no similarity to other code whose behaviour is duplicated (Koschke 2007). Copied code is typically considered harmful (Ducasse 1999). However Kasper et. al. (2006) argue that experimental variation, that is, a need to modify behaviour while maintaining stability, can be a valid reason for copying code and so not all code copying is necessarily bad.

In AWRA river modelling duplication is evident primarily as code redundancy and not as segments of copied code, i.e. there are multiple implementations that are semantically or functionally equivalent. Actual code copying in AWRA is rare. However the redundancy in AWRA shares many of the problems of code copying. Code copying increases maintenance cost because changes in one bit of functionality must be made in corresponding equivalent code. Furthermore the increased maintenance overhead introduces the risk that updates may not be correctly applied across the entire set of equivalent code. Juergens et al (2009) describe these costs and risks and find evidence for increased faults in copied code .The redundancy in AWRA code incurs an additional risk: code is not directly copied but rather re-implemented using differ technologies, therefore new mistakes may be introduced in attempting to build new, but equivalent, code. The resulting behaviour may not match the targeted behaviour in the original implementation. Furthermore, it is reasonable to expect that the proliferation of similar implementations in AWRA means a larger code base and thus in general increases the overall number of errors.

While the redundancies in AWRA fulfil multiple required uses cases they introduce a higher risk of errors across the AWRA systems. The need for a high quality AWRA software implementation meant verification of model implementations was critical during development of the AWRA system. We show that duplicate components across AWRA can be compared such that excepted behaviour of these components should be equivalent. Using this approach a verification process in AWRA identified bugs during software implementation. We found that duplication in a modelling system could be the basis of a comparative verification process useful during model development for finding errors in software structure, implementation, and specification across redundant duplicated components.

4. VERFICATION, VALIDATION, AND REDUNDANCY IN MODELLING

The verification of a simulation model is an activity that examines the software code and checks that this code matches the conceptual specification (Sargent 2005). Verification differs from validation of simulation models. Validation is the process of ascertaining that the model is accurate for the purpose and in the context it will be used (Law 2000). In the environmental modelling domain Refsgaard et. al. (2004) define code verification as the process of assuring that a software model is a true representation of a corresponding model but assert that verification is limited to the ranges within which it has been tested.

Many validation and verification approaches were used during the development of AWRA. Validation of river models (AWRA-R) assessed operational validity by comparing modelled outputs with observed data (Lerat 2011). Validation in AWRA uses approaches common in hydrology; for example split sample testing is used to provide assurance that calibrated parameters will produce reasonable results in operational contexts (Klemes 1986). Verification in AWRA included unit tests run continuously during development (Stenson 2011).

Software redundancies can be leveraged for verification through comparison of equivalent functionality. For example Freebairn et. al (2005) describe a process of porting a model from one modelling framework to another. They created a general purpose tool for verification of the ported software which mapped and compared equivalent outputs across implementations for simulations with equivalent inputs. Another example leveraging redundancy for verification arises in N-version programming, a methodology that intentionally produces more than one functionally equivalent program. An N-version execution environment executes all versions of the equivalent program and chooses the most consistent result (Avizienis 1995). Avizienis et al (1995) suggest that comparing different equivalent versions of N-version systems might be useful for identifying discrepancies during development as part of software verification. The duplication and redundancy across AWRA river modelling systems shares features of the software porting process described by Freebairn (2005) and of N-version systems described by Avizienis (1995). In all cases one or more systems or components are redundant. Thus the comparative approach for verification described for verifying a ported software model and suggested for verification of N-Version systems maybe applicable for AWRA.

5. COMPARATIVE VERIFICATION IN AWRA

5.1. Overview

A comparative code verification process was used as part of quality assurance in the development of version 3 of the AWRA River modelling system. The comparative process compared the behaviour of the operational and calibration river modelling systems. A key part of this process was an automated comparative test. The test checked that equivalent configurations of both systems produced equivalent results.

The development of AWRA river modelling in the calibration system preceded the development of equivalent functionality in the operational system. The calibration system was developed by a research team and functioned to facilitate: exploration of modelling concepts, rapid expansion and modification of model geographic coverage and as a means to produce calibrated parameter sets. In contrast the operational system used for production of data used in national water accounts, was developed by an implementation team to be fast, stable, and readily accessible and extensible. The specification for the operational system was defined through a combination of: the behaviour of the calibration system, defined operational requirements of the broader AWRA system, lightweight documentation, and informal communications between the research and implementation teams.

The river modelling comparative tests supported the simultaneous development of the operational and calibration river modelling systems. These tests compare the behaviour of the operational system to the calibration system and share many of the characteristics of the test tool Freebairn et. al. (2005) employed during software porting. In both cases the tests compared inputs to outputs in the newer system to equivalents in the existing system and this process was repeated as errors were identified and fixed.

5.2. Design and Development of Comparative Verification

The structure of the comparative tests and associated verification process closely relates to the structure of the AWRA river modelling system. AWRA-R represents rivers by modelling many connected rivers reaches. The overall river modelling system orchestrates the execution and connectivity of the individual river reach models and handles input/output and associated data pre and post processing (see Figure 1).

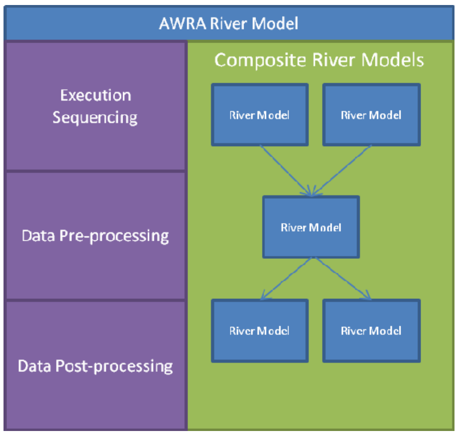
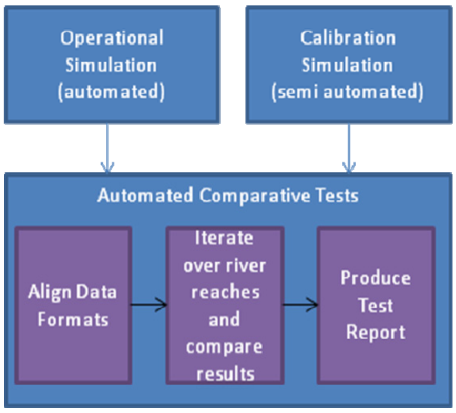


Figure 1. Model Structure

The comparative test system runs the operational system on a reach by reach basis and compares the outputs of each to a pre generated set of comparable outputs from the calibration system (see Figure 2.) The test generates a report that lists specific river reaches where modelled outflows differ significantly between the operational and calibration systems. In addition to one off manual execution the comparative tests were automated via a continuous integration system. Automating the tests helped ensure that subsequent development of the operational system, after initial testing, did not introduce bugs.


Operational Simulation (automated)
Calibration Simulation (semi automated)
Automated Comparative Tests
Align Data Formats
Iterate over river reaches and compare results
Produce Test Report
Figure 2. Comparative Tests

Directly comparing the operational and calibration AWRA systems was not straightforward. The specifications of the operational and calibration system differ slightly and therefore so does the expected behaviour. In particular input data that was identical in value between the systems was interpreted differently. Observed inflows at gauges defining the top of a river reach are used as inputs to a river reach model. Zero values in observed inflow time series are valid inputs in the calibration system. Difference like these make a test directly comparing results difficult. In the operational system, on the other hand, zero values are treated as an indication of invalid data and the system substitutes these with modelled values from upstream river reaches. In the AWRA system we solved this problem and others like it by developing a compatibility mode in the operational system. The compatibility mode sets the expected behaviour of the operational system to be the same as that of the calibration system.

The comparative test system formed an integral part of an iterative development process that included software developers working primarily on implementing the operational system and research hydrologists working on developing calibrated parameters, sourcing data, and specifying the overall conceptual model of the system. The implementations and calibration team members worked closely examining each case where the automated comparative test identified a mismatch between the behaviour of the operational and calibration systems. Typically this involved manual cross checking the inputs, outputs and states of a problematic river reach in both the operational and calibration systems to isolate the exact source of the discrepancy. In this way bugs were identified and fixed.

5.3. Results and Discussion

As the calibration system was considered the canonical reference implementation of the AWRA river model we expected that the comparative tests would primarily identify errors occurring during redevelopment of the operational system.

As expected the tests successfully identified bugs in the operational system. For example an error in the operational system occurred in interpretation of rainfall input data. The spatial averaging algorithm calculated a scaling factor by dividing the area of a contributing catchment with the sum of all contributing areas. Where there were no catchments associated with a particular river reach the algorithm was dividing by 0 producing a result that became an erroneous input to the core river model.

The comparative test system also successfully identified bugs in the implementation of the calibration system. For example, river reach models can be configured to model multiple outflows. That is they can model river anabranches where one river reach is split into two or more river reaches. A small number of river reaches require this kind of modelling. Configuration data files correctly specified anabranches but an error in calibration code meant this was being ignored during some calibration runs. In another similar case the tests identified an error in data management and interpretation. In some case data was not updated and become stale, compensating rules where developed in the calibration system but not specified. In this case comparative testing highlighted the need for data to be a "source of truth" for system configuration and minimising the use of hidden logic modifying configuration settings.

Finally the comparative test system also helped resolved ambiguities in the conceptual specification of the river model. For example, in the operational system irrigation diversions were used in headwater models and sometimes resulted in small differences in outputs compared to the calibration system. The differing results helped confirm that irrigation diversion modelling for headwaters should be switched on in the operational system but that the differing behaviour in the calibration system did not significantly alter modelled behaviour. That is a valid difference was identified.

Comparative verification will not detect some types of errors in code including: identical errors made in both systems, different errors across systems that result in the same output, and errors outside of the bounds of data and parameters used in tests. These kinds of errors can be identified through alternative complimentary verification and validation techniques. For example coupling more traditional approaches like unit testing provides additional quality assurance and avoids some of the issues with comparative verification. However comparative verification offers the advantage of a broader scope than individual unit tests, it can detect errors across the whole system. In the case of N-version programming Avizienis et. al. (1995) recognize significant cost in producing multiple software versions. In AWRA comparative verification avoids this cost as it leverages existing and incidental duplication between systems and therefore has a low additional development cost.

Comparative verification of AWRA version 3 encompassed approximately 600 river reach models. The verification process identified greater than 14 errors across the calibration and operational systems. Errors of a variety of types were identified including: incomplete data, algorithmic errors, errors in river reach connectivity algorithms and rules, parsing errors, data management issues, incorrect reporting, and use of incorrect code versions.

6. CONCLUSIONS

AWRA is a complex and heterogeneous modelling system consisting of many subsystems. These systems meet different use cases. There is significant conceptual overlap across the different implementations of AWRA. As a result there is some behavioural redundancy and duplication. This increases the risk of errors and thus the importance of verification of AWRA software and related processes and procedures. Duplication in AWRA river modelling occurred between the calibration system, used for exploring model concepts and finding optimal parameter sets, and the operational system, a readily usable, fast, stable, extensible, environment for producing outputs for national water accounts. We developed an effective comparative verification process for improving the quality of the operational and calibration river modelling systems. A compatibility mode was created such that the operational and calibration systems were expected to produce identical outputs. We produced a comparative verification process utilising the compatibility mode and leveraging redundancy across river modelling systems. This process identified inconsistencies between operational and calibration river modelling systems. We found that errors in software structure, specification, data management and interpretation, could be successfully identified and resolved through a comparative verification process harnessing redundancy and duplication present across a complex hydrological modelling system.

ACKNOWLEDGEMENTS

The authors would like to thank Julien Lerat, Avijeet Ramchurn, Adam Smith, and Andrew Frost for their contributions and the BoM CSIRO WIRADA Alliance for providing funding for this research.

REFERENCES

Avizienis, A. (1995). The methodology of n-version programming. *Software fault tolerance 3*, pp. 23-46.

Freebairn, A., J. Rahman, S. Seaton, J-M. Perraud, P. Hairsine and H. Hotham. (2005). Development of an automated testing tool for identifying discrepancies between model implementations. *MODSIM 2005 International Congress on Modelling and Simulation*. Modelling and Simulation Society of Australia and New Zealand, December 2005. Modelling and Simulation Society of Australia.

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

Juergens, E., Deissenboeck, F., Hummel, B., & Wagner, S. (2009). Do code clones matter?. *Proceedings of the 31st International Conference on Software Engineering*. pp. 485-495.. IEEE Computer Society.

Kapser, Cory, and Michael W. Godfrey. (2006). "Cloning considered harmful" considered harmful." *Proc. WCRE*. Vol. 6.

Klemeš, V. (1986). Operational testing of hydrological simulation models. *Hydrological Sciences Journal*, 31(1), 13-24.

Koschke, R. (2007). Survey of research on software clones. *Internat. Begegnungs-und Forschungszentrum für Informatik.*

Law, A. M. and Kelton, W. D. (2000). Simulation modeling and analysis. *McGraw-Hill series in industrial engineering and management science*. McGraw-Hill.

Lerat, J., Paydar, Z., Henderson, B., Stenson, M., and Van Dijk, A. I. J. M. (2011). Better use of prior information in the calibration of river system models. *MODSIM 11 International Congress on Modelling and Simulation*. Perth, Modelling and Simulation Society of Australia

Li, Z., Lu, S., Myagmar, S., and Zhou, Y. (2006). CP-Miner: Finding copy-paste and related bugs in large-scale software code. *Software Engineering, IEEE Transactions on*. 32(3), 176-192.

Penton, D., Leighton, B., Stenson, M., Rahman, J., and Bethune, M. (2011). Linking hydrological simulation models with workflow and optimisation software. *MODSIM 11 International Congress on Modelling and Simulation*. Perth, Modelling and Simulation Society of Australia

Refsgaard, J. C., and Henriksen, H. J. (2004). Modelling guidelines—terminology and guiding principles. *Advances in Water Resources*. 27(1), 71-82.

Sargent, R. G. (2005). Verification and validation of simulation models. *Proceedings of the 37th conference on Winter simulation*, WSC '05. Winter Simulation Conference. pp. 130-143.

Stenson, M. P., Fitch, P., Vleeshouwer, J., Frost, A. J., Qifeng, B., Lerat, J., Leighton, B., Knapp, S., Warren, G., Van Dijk, A., Bacon, D., Pena Arancibia, J., Manser, Paul., and Shoesmith, J. (2011). Operationalising the Australian Water Resources Assessment (AWRA) system. *Proceedings, Water Information Research and Development Alliance Science Symposium.*

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