

# A Model of Specification-Based Testing of Interactive Systems

Ian MacColl and David Carrington

Software Verification Research Centre,  
Department of Computer Science and Electrical Engineering,  
The University of Queensland, Brisbane QLD 4072, Australia  
[ianm@csee.uq.edu.au](mailto:ianm@csee.uq.edu.au)  
<http://www.csee.uq.edu.au/~ianm/>

In this paper we present a model of specification-based testing of interactive systems. This model provides the basis for a framework to guide such testing.

Interactive systems are traditionally decomposed into a functionality component and a user interface component; this distinction is termed dialogue separation and is the underlying basis for conceptual and architectural models of such systems. Correctness involves both proper behaviour of the user interface and proper computation by the underlying functionality. Specification-based testing is one method used to increase confidence in correctness, but it has had limited application to interactive system development to date.

Our starting point is the York model of interactors in which an interactive system (or component) is described in terms of functionality, presentation and interaction viewpoints, with the first two specified in a model-based notation and the last in a behaviour-based notation. Model-based specifications are represented as a state transition relation and behaviour-based specifications are represented as event sequences.

For a model-based notation, such as Z or Object-Z, specification-based testing is concerned with dividing the state transition relation, typically forming a partition of the input space of each operation. A single point in each equivalence class is used as a test input, and its projection onto the output space is a test oracle, used to determine the result of the test.

The state transition relation of a model-based specification (or testing information derived from it) can be interpreted as an implicit behavior-based specification. This is useful for test sequencing and to relate the testing information for each of the operations.

For a behaviour-based notation, testing information is derived from the specification to show an implementation conforms to the specification in terms of some relation, such as equivalence, quasi-equivalence or reduction for nondeterministic finite state machines, or equivalences based on trace or failure semantics for notations based on labelled transition systems (such as CCS and CSP).

We are using the model presented in this paper to develop a framework for specification-based testing of interactive systems. The framework aims to relate the model-based testing information, which is useful for unit testing, and the behaviour-based information, which is useful for system and usability testing. The paper is available at <http://www.csee.uq.edu.au/~ianm/model.ps.gz>.