OOPSLA'92
Vancouver, British Columbia, Canada
5 – 10 October 1992

Addendum
to the
Proceedings

Poster Submission—
Towards a Testing Methodology for Object-Oriented Systems

Report by:

Gail C. Murphy
Pok Wong
MPR Teltech Ltd.

Introduction

To date, interest in object-oriented technology has largely focused on the use of the technology at analysis, design, and implementation. An important part of the software life-cycle, however, is the testing required to ensure the software meets its specification. Most early proponents of object-oriented technology subscribed to the intuitive notion of the testing of object-oriented systems. That is, most believed that inherited features did not require retesting in the descendant class [CM90]. In [PK90], however, Perry and Kaiser showed that reuse of an inherited method in a subclass required testing in the new environment of the subclass even though the inherited tested method had not changed from its specification in the superclass.

This poster summarizes the issues involved in validating object-oriented software and proposes a methodology that will help organizations integrate appropriate testing into an iterative software process model. Tools have been developed that support the testing of Eiffel and C++ classes. These tools automatically apply test cases specified for superclass methods to subclasses.

The Challenges

A review of the literature on the testing of object-oriented systems identified only a handful of papers on the topic. Based on the reports in the literature and our own experience developing object-oriented systems, the following seven major challenges have been identified. A methodology has been proposed to meet these challenges.

Issue-1 How are testing activities integrated into an iterative software process model?

Issue-2 Given that many conventional strategies for guiding testing, like control-flow analysis, do not map to the decentralized nature of object-oriented software, what strategies are available to guide class and cluster[1] testing?

Issue-3 What constitutes sufficient testing for classes and clusters?

Issue-4 How can regression tests be formulated and conducted to ensure the integrity of classes modified under the open-closed principle [Mey88]?

[1] The term, *cluster*, is used to refer to a collection of classes that relate to a common aim.

Issue-5 What guidelines are available to the test designer who must account for state in the object?

Issue-6 How are abstract parent classes and generic classes tested?

Issue-7 Under what conditions must the features inherited from a paper be retested in the child class?

Testing and the Software Development Process

The object-oriented software testing methodology under development provides guidelines for testing the clusters and classes of software built using a statically typed object-oriented language. Cluster testing is specification-based, while class testing involves both specification and program-based testing of the methods of the class. Cluster and class level testing each involve two activities: 1) test case definition (including test data selection), and 2) test execution. For each cluster under development, the methodology proposes the following activities:

1. Cluster Analysis
2. Cluster Test Case and Test Data Definition
3. Class Design
4. Class Test Case and Test Data Definition
5. Implementation
6. Class Test Execution
7. Cluster Test Execution

Automated Testing Tools

An iterative software development process results in modifications to classes of already tested clusters. The types of modifications performed include changes to the inheritance hierarchy, changes to tested methods, additions of new methods, etc. To help support the regression testing of Eiffel and C++ classes developed within this iterative software development process, two tools, one for each object-oriented language, have been developed. These tools are extensions of the test program generator PGMGEN described in [Hof89].

The test tools verify an object-oriented class developed in Eiffel or C++ using test cases specified in a test script for the class. The tool generates a program in the appropriate language that automatically applies the test cases. For each test case, the trace (a set of messages to objects) for the test case is invoked and discrepancies between the expected and actual behavior are reported. The class under test is treated as a flattened class. That is, if the class has ancestors, the tool compiles and links the test scripts of the ancestors with the class under test and runs the resulting test drivers.

Conclusion and Future Work

Adoption of object-oriented technology for use in the development of highly reliable systems requires a testing methodology to complement existing object-oriented analysis techniques, object-oriented design techniques, and object-oriented programming environments. A methodology for testing object-oriented clusters and classes within an iterative software development process has been proposed. Tools have been developed to support the regression testing of classes developed in Eiffel and C++. The methodology and tools have been applied on a small-scale to two sample object-oriented clusters. In the future, we intend to apply the methodology and tools within a system development. The results of applying the approach will be used to drive further refinement of what constitutes adequate testing of classes, particularly when inheritance is involved.

References

[CM90] Thomas J. Cheatham and Lee Mellinger. Testing Object-Oriented Software Systems. In *Proceedings of the 1990 ACM Eighteenth Annual Computer Science Conference* Proceedings, pages 161–165. ACM, February 1990.

[Hof89] Daniel Hoffman. A Case Study in Module Testing. In *Proceedings of the Conference on Software Maintenance*, 1989.

[Mey88] Bertrand Meyer. *Object-Oriented Software Construction.* Prentice-Hall 1988.

[PK90] Dewayne E. Perry and Gail E. Kaiser. Adequate Testing and Object-oriented Programming. *Journal of Object-Oriented Programming*, pages 13–19, January/February 1990.

Contact information:

Gail C. Murphy
Pok Wong
MPR Teltech Ltd.
8999 Nelson Way
Burnaby B.C., Canada, V5A 4B5
E-mail: *murphy@mprgate.mprca*
wong@mprgate.mpr.ca