

What a large test suite!!  
I only wanted to test the  
the test cases that  
use BuggyClass1!!



Test Expert

# Ontology-based Unit Test-case Generation

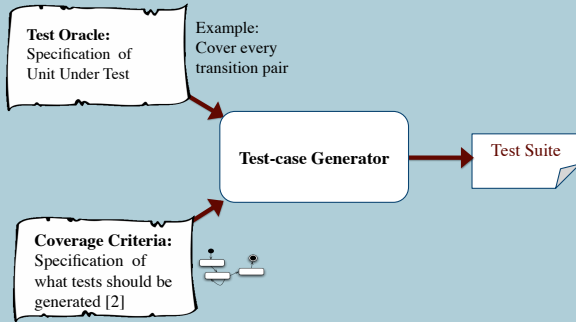
Valeh H. Nasser, Weichang Du, Dawn MacIsaac

Faculty of Computer Science, University of New Brunswick Fredericton, NB, Canada

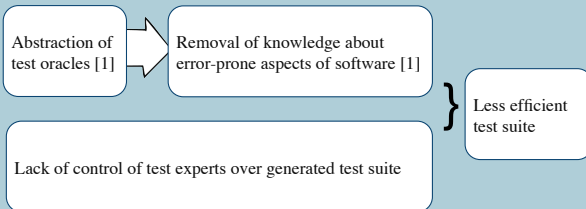


## Introduction

### A Test Generator System



### Problem

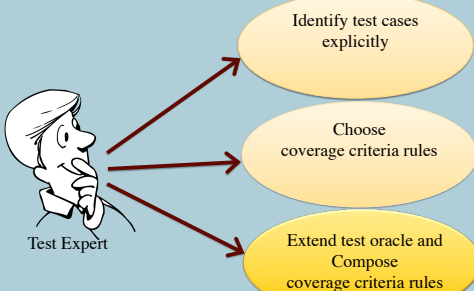


### Solution

The solution is to allow a test expert to:

- 1- Extend test oracle with their knowledge [1].
- 2- Define custom coverage criteria [1,3].

Test expert's control over the generated test suite:



## Ontology based Testing

### 1. Generate Test Structures:

A *test structure* denotes the structure of a single test case.

Model the test oracle in an ontology and extend it with expert knowledge.

Define coverage criteria rules:

criteria for selection of test structure based on test oracle and expert knowledge -> structure of a test case

Generate test structures using reasoning.

### 2. Check Redundancy of a Test Structure:

Define redundancy checking rules:

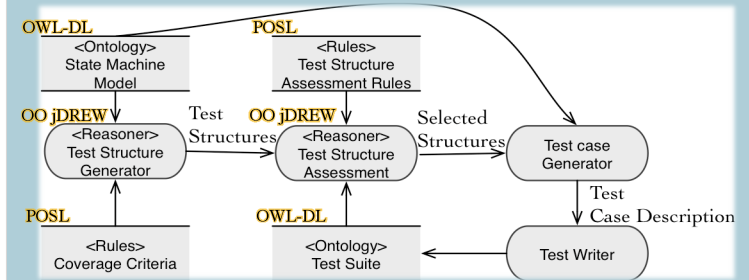
specification of a test structure based on the test suite ontology -> existence of a test case

Use reasoning to Identify existence of a test with a given test structure in a partially generated test suite ontology.

### 3. Generate Test-cases:

Generate test cases for a test structure that is not satisfied by the test suite and add it to the test suite ontology.

## System Architecture



## Concluding Remarks

Ontology based representation of test oracle is extensible and empowers test experts to use their knowledge and define custom coverage criteria to generate efficient test suites.

## Bibliography

- [1] S. Benz. Combining test case generation for component and integration testing. In Proceedings of the 3rd international workshop on Advances in model-based testing, pages 23–33. ACM Press New York, NY, USA, 2007.
- [2] H. Zhu, P. Hall, and J. May. Software unit test coverage and adequacy. ACM Computing Surveys (CSUR), 29(4):366–427, 1997.
- [3] G. Friedman, A. Hartman, K. Nagin, and T. Shiran. Projected state machine coverage for software testing. In Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis, pages 134–143. ACM New York, NY, USA, 2002.
- [4] M. Ball. OO jDREW: Design and Implementation of a Reasoning Engine for the Semantic Web. Technical report, Technical report, Faculty of Computer Science, University of New Brunswick, 2005.
- [5] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, L. Stein, et al. OWL Web Ontology Language Reference. W3C Recommendation, 10:2006–01, 2004.
- [6] H. Boley. POSL: An Integrated Positional-Slotted Language for Semantic Web Knowledge. <http://www.ruleml.org/submission/ruleml-shortation.html>, 2004.

For more information, please refer to our paper titled “Ontology-based unit test-case generation” in Proceedings of 2009 UNB CS ResearchExpo.  
Contact information: valeh.h@unb.ca.