# A Model-Driven Approach to Web Application Testing

Jairo Pava, Yanelis Hernandez, Tariq M. King, Peter J. Clarke
Florida International University, Miami, FL 33199

SRAI
The Honors College @
Florida International University
Student Research & Artistic Initiatives

THE HONORS COLLEGE
Florida International University
Excellence with Honors

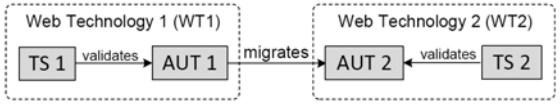
## I. Abstract

The growing number of web applications used in business today has come as a result of the many advances in web-based technologies. As the demand for flexibility, mobility, and connectivity to the Internet pushes the development for more complex web applications, the testing frameworks used to test new software struggle to keep pace. Therefore, the rapid evolution of web-based technologies becomes a problem for businesses that require constant and thorough testing of their web applications. Using a model driven approach, we can create a meta-model to support testing web-based technologies. By generating a meta-model for the constructs in web applications and a meta-model for the testing frameworks, we can simplify the transition between one form of technology and another in terms of testing. The meta-model can then be used to create a generic testing framework for any new type of web application technology, thereby saving a business time, money, and complexity.

## II. Background

- Web Technologies are the building blocks of programs designed to be used through a web browser via the Internet. Examples of applications built using these technologies include e-commerce web sites, online banking, stock exchanges, and much more.
- Regression Testing refers to re-running tests after an application has been altered to determine whether or not new errors have been introduced into previously tested components.
- Model-Driven approaches involve the use of software models and transformations between these models for building complete applications. Models can be made platform independent so that they do not rely on any technological constructs, and later ported to the target platform.

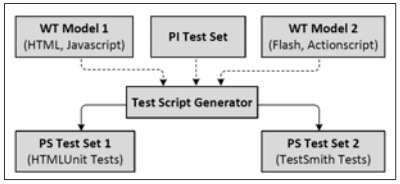
## III. Testing Web Applications



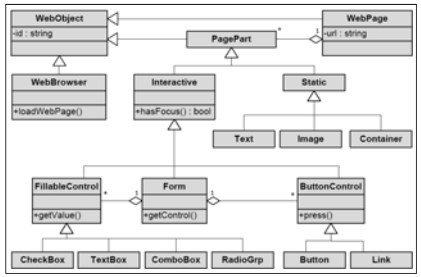

- Businesses automate testing by developing a test script (TS1). This script is read by a testing tool to validate their web application under test (AUT1).
- Migration of the web application involves using a new set of web technologies for building a new version of the application, thereby making TS1 unusable.
- A new test script (TS2) would therefore have to be created to validate the updated application under test (AUT2), which involves costly re-work.

## IV. Model-Driven Approach

- Using a model-driven approach we develop a platform independent model for the constructs used in testing a particular set of web technologies.

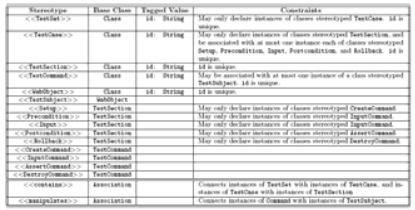



- Web technology models are designed to represent the constructs of particular web applications. WT1 models the constructs for HTML and JavaScript, while WT2 models the constructs used for designing Flash and Actionscript web applications.
- The models are combined to generate specific testing scripts relevant to the chosen web technology. WT1 is combined with the PI test set to generate a testing script in the test script generator to validate HTML and JavaScript web applications. If the application migrates to Flash and Actionsript, the test set can also migrate to automatically generate a new testing script by combining WT2 and the PI test set.
- A conceptual model for the user interface web application provides abstractions for the visual elements in the application that are relevant to testing.




- WebObject is a general representation for any element of the web interface. These include web pages and any other elements contained within them.
- WebPages are composed of interactive and static page parts
- Static parts remain fixed despite external stimuli, e.g., text.
- Interactive parts dynamically interact with a user, e.g., forms.

- UML Profiles specify the stereotypes, base classes, tagged values, and constraints of the conceptual model above



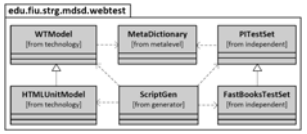
| Stereotype | Base Class | Tagged Value | Constraints |
|---|---|---|---|
| <<TestSet>> | Class | id : String | May only declare instances of classes stereotyped TestCase. id is unique. |
| <<TestCase>> | Class | id : String | May only declare instances of classes stereotyped TestSection, and be associated with at most one instance each of classes stereotyped Setup, Precondition, Input, Postcondition, and RollBack. id is unique. |
| <<TestSection>> | Class | id : String | id is unique. |
| <<TestCommand>> | Class | id : String | May be associated with at most one instance of a class stereotyped TestSubject. id is unique. |
| <<webObject>> | Class | id : String | id is unique. |
| <<TestSubject>> | WebObject | | |
| <<Setup>> | TestSection | | May only declare instances of classes stereotyped CreateCommand. |
| <<Precondition>> | TestSection | | May only declare instances of classes stereotyped InputCommand. |
| <<Input>> | TestSection | | May only declare instances of classes stereotyped InputCommand. |
| <<Postcondition>> | TestSection | | May only declare instances of classes stereotyped AssertCommand. |
| <<RollBack>> | TestSection | | May only declare instances of classes stereotyped DestroyCommand. |
| <<CreateCommand>> | TestCommand | | |
| <<InputCommand>> | TestCommand | | |
| <<AssertCommand>> | TestCommand | | |
| <<DestroyCommand>> | TestCommand | | |
| <<contains>> | Association | | Connects instances of TestSet with instances of TestCase, and instances of TestCase with instances of TestSection |
| <<manipulates>> | Association | | Connects instances of Command with instances of TestSubject. |

- Stereotypes represent specific meta-classes
- Base Class denotes an extension relationship from a UML meta-class or inheritance from another stereotype
- Tagged Values define attributes of the stereotype
- Constraints enforce restrictions on how the meta-model may be used

## V. Case Study

- Two versions of the same web application are designed to simulate the migration of a business from one web platform to another that uses technologies not supported by the current testing tool.

- First version developed using HTML 4.01 and JavaScript
- Second version developed using Flash 9 and ActionScript
- A prototype of the model-driven solution was implemented using Java 5.0
- An initial test set of twelve test cases was designed to test both applications for functionality, glitches, and stability.




- The webtest package shows the main classes from the sub-packages of the prototype test implementation
- MetaDictionary stores the different types from a conceptual web interface model and is then used to model the web testing technology WTModel and platform independent test set PITestSet.
- ScriptGen iterates through FastBooksTestSet and generates a testing script based on specific constructs of the HTMLUnitModel. Abstract test commands are translated by ScriptGen into a testing script that is used to test HTML web applications.

## VI. Discussion

- The purpose of the work was successful in that all of the test cases developed for the case study were effectively translated into HTMLUnit tests. This shows that the abstract constructs used in the prototype were sufficient to automatically generate testing scripts that are capable of testing a web application using a model-driven approach. The current model does not implement the technology model for Flash, but this could be incorporated by extending the generalized classes in the model.
- Future work includes expanding the model to include a wider coverage of web controls and widgets. Also, a model-driven architecture tool such as Eclipse Modeling Framework can be used to automatically detect model constraint violations.