

Web Application Testing

Yao Wu

1

2 Papers

- “Analysis and Testing of Web applications”
by Ricca and Tonella
- “Going Faster: testing the web application”
by Hieatt and Mee

2



Analysis and Testing of Web Application

- Motivation
- Contributions
 - a UML model of Web Applications
 - Tools: ReWeb & WebTest
- Experiments
 - Apply tools to Wordnet and amazon
- Conclusion

3



Motivation

- Evolve rapidly and undergo frequent modification.
- Iterative development process models.
- Analysis model
- Develop tools

4

Meta Model

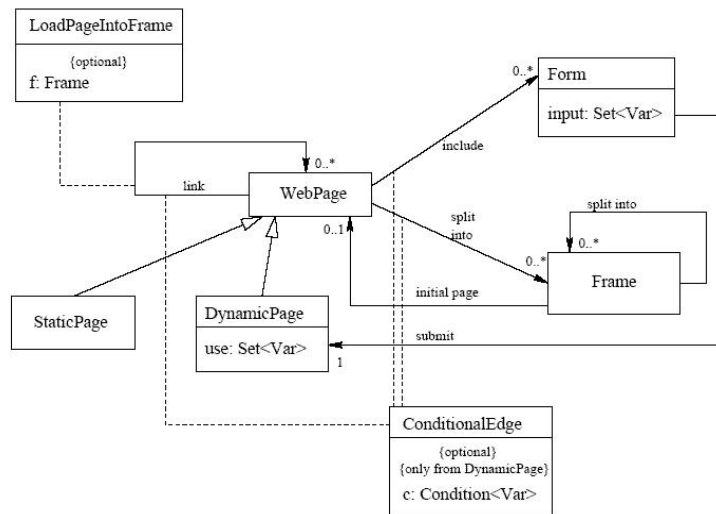


Figure 1: Meta model of a generic Web application structure. The model of a given site is an instantiation of it.

Analysis Model

- An example of tiny [website](#) (model with frame)

Model with frame

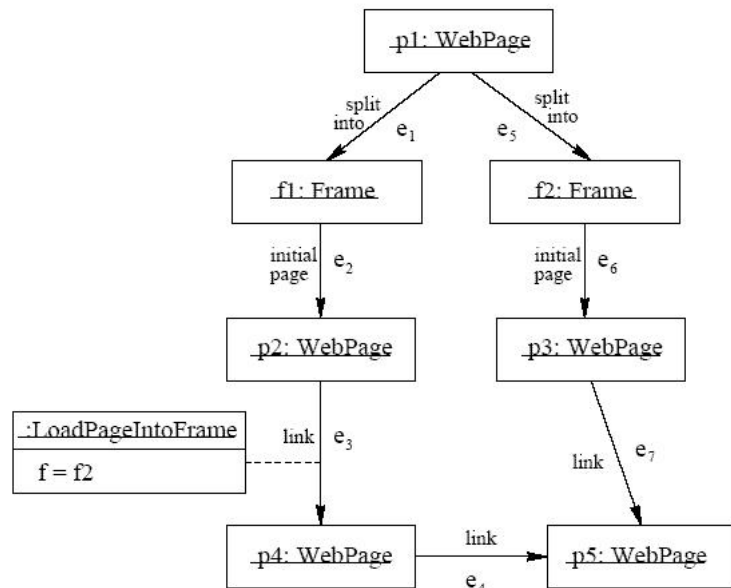


Figure 2: Example of model with frames.

Model with Form

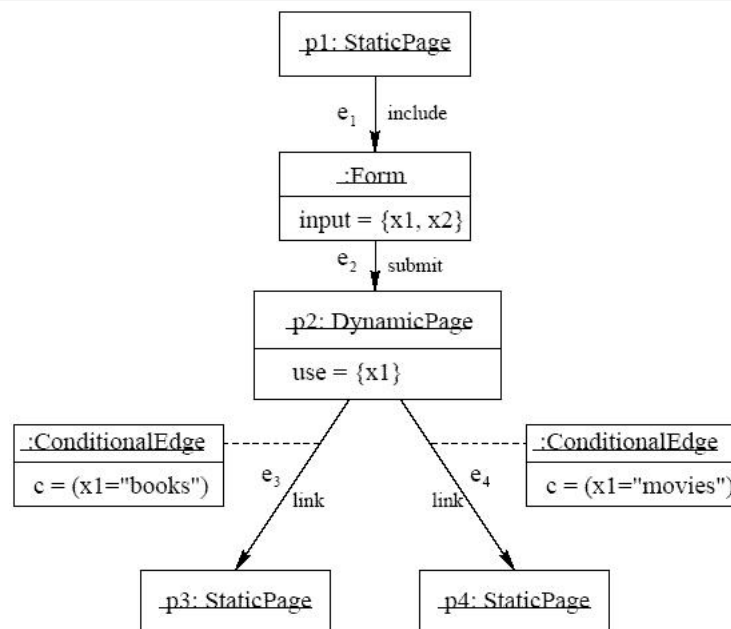


Figure 3: Example of model with a form.



Testing: Static Verification

- Inspections,
- Static analyzer scan HTML pages in a website and detect faults.
- Models can help determine
 - Unreachable pages
 - Ghost pages
 - Reaching frames
 - Data dependences
 - Dominators
 - Shortest Path

9



Testing: Dynamic Validation

- White box testing: test suite, test criterion.
- Test case and its execution.
- Some white box testing criteria
 - Page testing
 - Hyperlink testing
 - Def-use testing
 - All-uses testing
 - All-paths testing
 - Loop k-limiting and path independence

10



Testing: Dynamic Validation (cont...)

- Static pages not containing forms can be disregarded.
- Reduced Graph
 - static page without forms removed
 - All its predecessors are linked to all its successors
 - Add a fictitious entry node and exit node

11



Test Case Generation

- Test suite: a set of paths. Automated.
- Path Expression
 - Variable: edge label
 - Operator: + *
 - examples
- Website --> path expression
- Path expression --> test cases
- Test engineer insert input values
- Execution, download output pages

12

ReWeb and TestWeb Tools

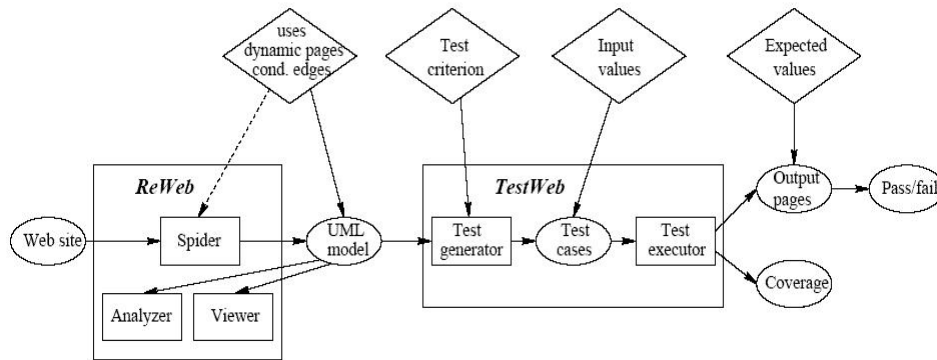


Figure 4: Roles of **ReWeb** and **TestWeb** in the analysis and testing process.

13

ReWeb and TestWeb Tools

- ReWeb consists of 3 Modules:
 - Spider
 - Analyzer
 - Viewer
- TestWeb contains:
 - Test case generation engine
 - Test executor
- Test engineer assesses pass/fail of each test case.

14

Experiment Result | Wordnet

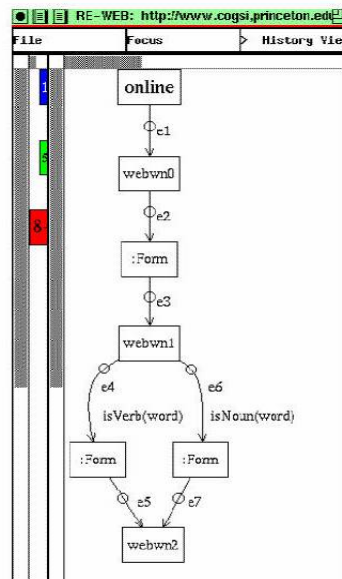


Figure 6: Portion of UML model of the Web application 5 Wordnet, with state unrolled.

Experiment Result - amazon

- A defect:
2 sequence of pages:
 - gift, shipping, shipping, payment
 - Shipping, shipping, gift, payment



Conclusion

- Provide good understanding of site organization
- Good coverage
- Semi-automatic and requires user input

17



Going Faster: Testing the Web Application

- Motivation
- Approaches – testing first
- Test challenges and solutions with testing first approach.
- Conclusion

18



Motivation

- Testing is often last in developer's minds when pressured to deliver something.
- Resulting code is often unstable, fragile, difficult to adapted to new requirements.
- A radical approach: centered on testing.

19



Approaches – testing first

- Testing first: Write a unit test for a new piece of functionality before the new code is written.
- Benefit:
 - Pervasive quality of testability throughout the code base.
 - Code reusable.

20



An Example –Java servlet

- Unit testing helps removing dependence on a particular technology.
- Traditional way: override service method.
Problem: servlets are awkward to test.
- Test-first way: build a new class for code that handles requests. In service method, just create an object of new class.
Adv: have a testable design, more appropriate division of labor, and more tech-independent.

21



How to make use of bugs?

- Bug count – help pinpoint areas that need more testing.
- Bug – write a test that “catches” the bug, fill a hole in test suite.
- Tracking bugs – something seriously wrong exists with test suite. Assign to a pair of programmers.
- Constantly growing test suite, to cover as much of the application as possible.

22



Use tests to document the system

- Tests are written first, they completely define what the code should do.
- Quicker to learn code.

23



Testing Challenges and Solutions

- Problem 1: difficult to test both client and sever at once. Don't want to split tests.
- Solution:
 1. Test parts of sever that are not directly concerned with being part of application.
 2. Test parts of client side that have no server interaction.
 3. Write functional tests of a low grain in the server-side language that simulate the request/response.

24



Testing Challenges and Solutions, (cont...)

- Problem 2: Testing data in the output, not testing the HTML web page.
- Solution:
 1. Generate XML output-pure data from domain objects.
 2. Transform XML into HTML.
- XML is easily testable.

25



Conclusion

- Focus on tests and testability helped all projects to be faster.

26



Reference

- “Analysis and Testing of Web applications”
- “Going Faster: testing the web application”