Modeling and Testing Web-based Applications

Jeff Offutt
George Mason University
Joint work with Ye Wu

Thesis Statement

- Web applications are heterogeneous, dynamic and must satisfy very high quality attributes
- Use of the Web is hindered by low quality Web sites and applications
- Web applications need to be built better and tested more
Research in a Nutshell

- Web applications control flow cannot be determined statically
- Define *atomic sections* that have structural information and dynamic content
- Model is composition of atomic sections and *regular expressions*
- Model is used to support design, testing, and maintenance

Multi-tiered Web Software Systems

Client-server ... 3-tier ... N-tier ...
Important Web Software Quality Attributes

1. Reliability
2. Usability
3. Security
4. Availability
5. Scalability
6. Maintainability
7. Performance & Time-to-market

Customers have little “site loyalty” and will switch quickly, thus time to market is much less important than in other application areas. (but still important!)

Based on an informal survey of a dozen software development managers, 2000.

New Essential Problems of Web Site Software

1. Web software content providers and services offer dynamically changing flow of control
   - Web pages are created by software on user request
   - The UI controls vary depending on state: the user, previous choices, server-side data, even time of day
   - Examples: amazon.com, netflix.com, aol.com, washingtonpost.com

2. Web software is extremely loosely coupled
   - Coupled through the Internet – separated by space
   - Coupled to diverse hardware and software
   - Web services dynamically couple with other services after deployment – without human intervention!
Problem 1: Dynamic Flow of Control

How can we ensure the reliability of this type of system?

Problem 2: Loosely Coupled

How can we ensure the reliability of this type of system?
Modeling Dynamic Control Flow

- Dynamic web pages are created when users make requests
- The program the user interacts with changes dynamically
- Unlike traditional software, we cannot determine potential flows of control before execution
- But all the pieces of the web pages and the programs are contained in the software …

Atomic Sections

HTML with static structure and content variables

```
P1 = PrintWriter out = response.getWriter();
    out.println("<HTML>")
    out.println("<HEAD><TITLE>" + title + "</TITLE></HEAD>")
    out.println("<BODY>")
    for (int i=0; i<myVector.size(); i++)
        if (myVector.elementAt(i).size > 10)
            out.println("<p><b>" + myVector.elementAt(i) + "</b></p>"));
        else
            out.println("<P>" + myVector.elementAt(i) + "</p>"));
    out.println("</BODY></HTML>");
out.close();
```

```
P2 = for (int i=0; i<myVector.size(); i++)
    if (myVector.elementAt(i).size > 10)
        out.println("<p><b>" + myVector.elementAt(i) + "</b></p>"));
    else
        out.println("<P>" + myVector.elementAt(i) + "</p>"));
```

```
P3 = out.println("<P>" + myVector.elementAt(i) + "</P>"));
```

```
P4 = out.println("</BODY></HTML>");
out.close();
```

```
Web Service Application Model

- Web pages are modeled as composite sections and regular expressions
  - Sequence, selection, aggregation
  - Conditions on inclusion of composite sections
- The previous example produces:
  \[ p \rightarrow p_1 \cdot (p_2 \mid p_3)^* \cdot p_4 \]
- Models can be produced automatically

Test Criteria

- Tests are created by deriving sequences of transitions among the web software components and composite sections:
  - All productions in the grammar (each atomic section)
  - Each selection used once
  - Each possible aggregation
  - MC/DC type coverage of conditions on productions
Testing Loose Couplings

- Current integration testing covers couplings among the software components – requiring source
- Essential part of integration is how software components communicate
- XML stores data independently of type, without regard to format, in arbitrary order
- Tests created by modifying existing XML messages according to the XML Schema

Benefit to AOL

- AOL is a content provider
- Model can support AOL, open source products, and content producers
- Better design, maintenance and testing can lead to better quality products and more satisfied customers
- Web applications can be more accessible to more people
Contact Information

Jeff Offutt
Information & Software Engineering
George Mason University
+1 703-993-1654
ofut@ise.gmu.edu
www.ise.gmu.edu/~ofut/
Ye Wu, wuye@ise.gmu.edu