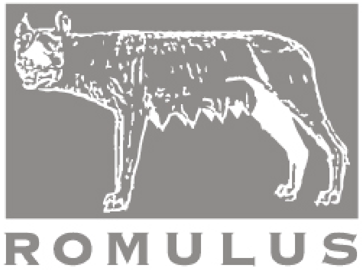
IADIS International Conference WWW/Internet 2009

Functional Testing Based On Web Navigation With Contracts

Boni García, Juan C. Dueñas, Hugo A. Parada

Universidad Politécnica de Madrid



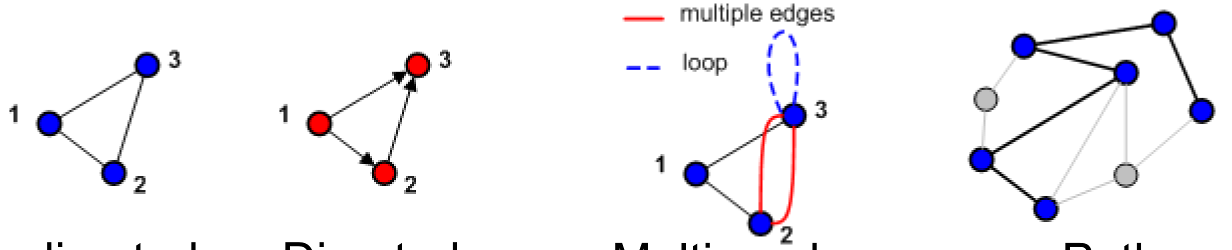
Introduction

- **Web** applications are more and more sophisticated
  - Testing is the main technique to ensure quality
- **System testing**: important but complex and costly
  - Automatic testing could save time and effort
- **Functional testing**: key to ensure external quality
  - Need to specify requirements: contracts
- Problem at hand: **automatic system testing** based on **contracts** for **web** applications with Java in the server-side and **open-source** technologies

Table of Contents

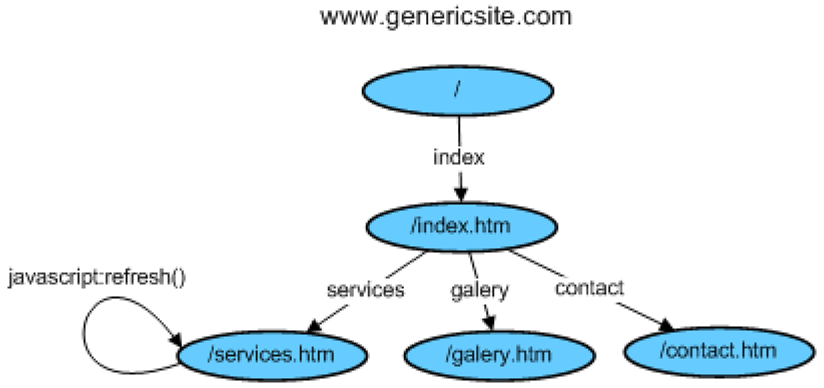
Background – Graph Theory

- Graph = set of nodes connected by links



Undirected Directed Multigraph Path

- A web site can be modeled as a **finite multidigraph**



- Open-source graph Java library:
  - JUNG v2.0 : http://jung.sourceforge.net

Background – Contracts

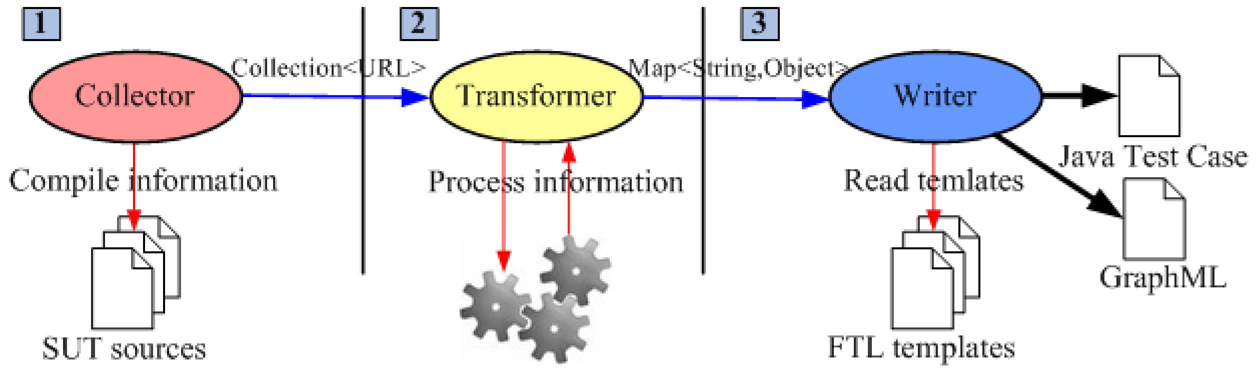
- Liu's Method: specify the functionality of a software component by means of pre and post-conditions
- Programming-by-contract approach
  - Pre/Post-condition: condition that must be met just before/after the execution of a portion of code
  - Invariant: condition whose value doesn't change during the execution of a portion of code
- Open-source contract Java library:
  - **Contract4J5** : http://www.contract4j.org

Background –Testing Tools

- We need a tool to test complete web applications, just like a browser does
- Headless web browser = GUI-less browser
- Open-source headless web browsers for Java:
  - **Selenium :** http://seleniumhq.org
    - In silent-mode.
  - **HtmlUnit :** http://htmlunit.sourceforge.net
    - Better JavaScript support than HttpUnit

Table of Contents

Method – Methodology



- Collector: Compiles structure and contract
- Transformer: Processes information
- Writer: Generates Java test cases (JUnit v3/v4, TestNG) and graph (GraphML)
  - FreeMarker Temaplates: http://freemarker.org/

Method – Algorithm

- Algorithm to decompose a multidigraph into non-hamiltonian paths. Restrictions:
  1. Each node/link must be visited at least once
  2. First node will be the home page
  3. When reaching a leaf node we start a new path from the beginning
  4. Loops will have priority while browsing, for reducing the number of paths
  5. Browsing will finish when all the links are visited at least once

Method – Implementation

- The method has been fully implement in the Automatic Testing Platform (ATP) tool

  http://www.ict-romulus.eu/web/atp4romulus

- The web targets are based on the Roma Metaframework

  http://www.romaframework.org

Table of Contents

Conclusions

- We have created a grey-box testing approach, based on the structure (white-box testing) and the specification in form of contracts (black-box testing)
- It is based on open-source technologies: JUNG, GraphML, Contract4J, Selenium, HtmlUnit, FreeMarker, JUnit, TestNG
- Fully implemented in ATP, the testing tool for Roma Metaframework based web applications

Conclusions – Future Work

- Validation of the proposed method
  - Using different graphs/webs
- Extension of the method to another web frameworks
  - For example, Spring Web Flow

SEVENTH FRAMEWORK PROGRAMME

Thank you

Boni García

Universidad Politécnica de Madrid

bgarcia@dit.upm.es

ROMULUS