Models and Techniques for the Reverse Engineering of Web Applications

Candidato
Domenico Amalfitano

Tutor
Prof. Anna Rita Fasolino

Co-Tutor
Ing. Porfirio Tramontana
Research Activity description

- **Context**
  - Rich Internet Applications (RIA)

- **Objectives**
  - Defining models to describe the characteristics of a RIA.
  - Proposing Reverse Engineering techniques for obtaining these models from existing RIAs.
  - Validating the proposed models and techniques by experiments.
Introduction to RIA

- A new generation of Web applications.
  - Almost all the business logic of the application is on the Client.
  - Client and Server exchange few amount of data (URL of the RIA doesn’t change).
  - The communication among Client and Server can be asynchronous.
- More dynamic, interactive, responsive and usable than traditional ones, with a rich user interface.
- They overcome the click-and-wait paradigm of the traditional Web applications.
- AJAX and Flex are the most used technologies to develop RIA.
- Google Maps is a typical RIA example.
State of the Art

- New challenges for the Software Engineering due to the birth of the RIA applications.
  - Models, methods, techniques and processes used to develop the traditional Web application, are not suitable for the RIA.
    - At least rearranged
  - The absence of appropriate models and development processes produce poorly documented RIAs.
    - Some tasks become harder:
      - Comprehension
      - Automated Test
      - Maintenance & Reengineering

- Considerable effort in scientific community devoted to the proposal of models suitable in processes of Forward\(^1\) and Reverse Engineering\(^2\).

---

1. “Necessity of methodologies to model Rich Internet Applications”.
   “Designing Rich Internet Applications with Web Engineering Methodologies”.
   “An Architectural Style for Ajax”.
2. “Migrating Multi-page Web Applications to Single-page AJAX Interfaces”
   “Crawling AJAX by Inferring User Interface State Changes”
   “State-Based Testing of Ajax Web Applications”
The Behavioural Model of a RIA

- Goal:
  - Finding suitable models to document the behaviour of the RIA.
- RIAs usually offer complex GUIs to their users.
- Focus on modeling the behaviour of the GUI.
  - Classic models adopted for GUI descriptions are Event Flow Graphs and Finite State Machines.
  - Finite State Machines models were adopted
A proposal: an FSM Model of the RIA GUI

- FSM represents all the elaboration states where the RIA receives any input solicitation by its user.
- Each state of the RIA is described by the client Interface shown to the user at the interaction time.
- Each client interface is characterized only by a sub-set of its widgets.
- Transitions are associated with user interactions that trigger the RIA migration towards a new state.

The UML class diagram showing the information characterizing the proposed FSM model of an RIA
FSM - an example of transition

Reconstructing the FSM model by Reverse Engineering Techniques

- Need for a Reverse Engineering process to reconstruct the FSM
- Two possible approaches for Reverse Engineering:
  - Static Analysis
    - Not sufficient, since RIAs have no stable source code, on the client side.
  - Dynamic Analysis
    - Based on trace execution collection.
    - At run time, data related to browsed interfaces and fired events are extracted and used to reconstruct the behavioural model of the RIA.
- Open problems:
  - State Explosion problem.
  - Termination problem.
A ‘Waterfall’ Reverse Engineering Process

- **Execution Trace Collection.**
- **Trace Analysis:** heuristic classification criteria are used to solve the data explosion problem, by clustering similar interfaces and events recorded during the Execution Trace Collection.
- **FSM model Abstraction:** an expert analyses and validates the clusters proposed by the criteria, assigns each of them with a logic interpretation, and finally obtains a FSM of the application.

- Limitations:
  - The FSM is obtained only at the end of all three steps.
  - The FSM Model Abstraction step relies on a costly and human intensive validation activity.
  - Difficult to apply a termination criteria.

A Validation Experiment

- **Goals of the experiment:**
  - assessing the technique effectiveness
  - comparing the effectiveness of different interface equivalence criteria

- **Experimental Materials**: (four involved RIAs)
  - W1: http://app.ess.ch/tudu/welcome.action
  - W2: http://www.pikipimp.com
  - W4: http://www.buttonator.com

<table>
<thead>
<tr>
<th>Subject Application</th>
<th>Use Cases</th>
<th>Scenarios</th>
<th>Gold Standard States</th>
<th>Gold Standard Transitions</th>
<th>Collected User Session Traces</th>
<th>Collected Interfaces</th>
<th>Best Criterion</th>
<th>Edit Distance</th>
<th>Correct Interface Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>8</td>
<td>17</td>
<td>15</td>
<td>52</td>
<td>30</td>
<td>1885</td>
<td>C3</td>
<td>14</td>
<td>85%</td>
</tr>
<tr>
<td>W2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>16</td>
<td>8</td>
<td>533</td>
<td>C3</td>
<td>8</td>
<td>65%</td>
</tr>
<tr>
<td>W3</td>
<td>3</td>
<td>10</td>
<td>4</td>
<td>9</td>
<td>11</td>
<td>731</td>
<td>C3</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>W4</td>
<td>1</td>
<td>8</td>
<td>19</td>
<td>54</td>
<td>11</td>
<td>829</td>
<td>C2</td>
<td>0</td>
<td>100%</td>
</tr>
</tbody>
</table>

Iterative Reverse Engineering Process

- A more ‘Agile’ approach based on an iterative process
  - It supports the *incremental* reconstruction of the FSM
  - The *feedback* about the reduction techniques is provided at each process iteration
  - The process ends on the basis of a Termination Criterion:
    - RIA Coverage Criterion
    - Effort devoted to the whole process.
The CReRIA environment

- CReRIA is the integrated environment for dynamic analysis designed for supporting the proposed RIA comprehension process.

- Functionalities:
  - Provides a Web browser for navigating the RIA and performing user sessions tracing;
  - extraction and recording of information about traced user sessions, such as user interfaces and events that occurred during the navigation;
  - classification of interfaces and events according to different abstraction criteria;
  - supporting the Concept Assignment task on the basis of information collected or abstracted in the previous steps of the process.

- Developed in Java language supported by a MySQL DBMS
The CRerRIA GUI

the tool is available at http://wpage.unina.it/ptramont/downloads.
Case Studies

- Experimental question: Assessing the improved efficiency of the proposed comprehension process with respect to a traditional waterfall one.
- Evaluation: Two metrics that express the actual utility of the classification suggestions provided by the tool during the process
  - Accepted State Suggestion Ratio (ASR-S)
  - Accepted Transition Suggestion Ratio (ASR-T)
- Experimental materials: three RIAs that were explored in the CReRIA environment

<table>
<thead>
<tr>
<th></th>
<th>W1</th>
<th>W2</th>
<th>W3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Proposed FSM transition #</td>
<td>Accepted Suggestion #</td>
<td>ASR-T</td>
</tr>
<tr>
<td>T1</td>
<td>28</td>
<td>304</td>
<td>50%</td>
</tr>
<tr>
<td>T2</td>
<td>34</td>
<td>426</td>
<td>70%</td>
</tr>
<tr>
<td>T3</td>
<td>47</td>
<td>518</td>
<td>85%</td>
</tr>
<tr>
<td>TBest</td>
<td>47</td>
<td>518</td>
<td>85%</td>
</tr>
</tbody>
</table>
Example of FSM Model

- The FSM Model of TuDu RIA

Conclusions & Future Works

○ Conclusions
  ● A proposal of a FSM based model of the GUI behaviour
  ● Two reverse engineering approaches for reconstructing it, which were validated by experiments

○ Future works
  ● Defining and validating additional heuristic clustering criteria
  ● Applying the proposed RE approaches in different scenarios, such as:
    ○ Automatic testing of RIA
    ○ Maintenance of RIA functionalities
    ○ Reengineering of a RIA
    ○ RIA Crawling.