Web Pages Classification using Concept Analysis

Porfirio Tramontana
Anna Rita Fasolino
Dipartimento di Informatica e Sistemistica
University of Naples Federico II, Italy

Giuseppe A. Di Lucca
RCOST – Research Centre on Software Technology
University of Sannio, Benevento, Italy
The Context

- With the diffusion of new paradigms and technological solutions for the Web (RIA and Ajax, SOA, …), existing Web Applications are rapidly become legacy

- A strategic objective: integrating that existing applications with the new platforms

- An open issue: defining effective approaches for analysing and classifying the Web Applications User Interface
Automatic Web Pages Classification

- Web Applications interact with users by means of Client Pages that are dynamically generated.

- Given a classification of Built Client pages in a set of equivalent classes:
  - … classes may correspond to the different reached Web Application scenarios
  - … classes may correspond to the different results of the execution of test cases

- The problem:
  - To propose a technique for the automatic identification of the equivalence class a Web page belongs to.
A possible application scenario

- Designing wrappers that encapsulate the original UI with the aim of exporting a renewed interface, such as a Web Service one
  - The possible reached scenarios represent the classes to recognise.
  - The Wrapper has to automatically identify the class in order to know the state of the interaction with the UI and the actions to be performed
The proposed solution

- To find, for any class, a combination of Web page features for any class that are satisfied just by pages belonging to that class (Classification model)

- We are interested to a solution that is:
  - Accurate
    - The classification approach should be very reliable, in order to support automatic tasks
  - Efficient
    - The classification approach should support a fast classification of dynamic built client pages
The proposed approach

Three step iterative process:

1. Feature Candidature
   - Generation of a set of features, representing page characteristics

2. Feature Reduction
   - Proposition of a combination of the candidate features allowing the identification of the class of any Web page

3. Feature Validation
   - Evaluation of the accurateness of the proposed classification model (i.e. the set of discriminating expressions)

The resulting classification model allows a Classifier module to automatically determine the class any Web page belongs to.
1 - Feature Candidature

- A feature is a characteristic retrievable in a Web page interface
  - It is retrievable by analysing its HTML code
  - A feature can be associated to a XPath query (if the HTML code is XHTML-compatible)

- A set of features that could be useful for discriminating a class are proposed
  - Ideally speaking, we search for features that are retrievable just in the pages belonging to a class

```
//a/text()="Activities"
```

```
//h1/img
```

```
/html/body/div[4]/form
```

---

2 - Feature Reduction

- The goal of the Feature Reduction step is to obtain a **Discriminating Expression** for each class, i.e. a combination of features that is true for each Web page belonging to the class, false elsewhere.

- The Feature Reduction is executed on the set of candidate features.
  - A training set of pre-classified pages is adopted.
## Evaluation Matrix

**Candidate Features**

<table>
<thead>
<tr>
<th></th>
<th>Login</th>
<th>Logout</th>
<th>Incorrect</th>
<th>AdminLog</th>
<th>LinkToHome</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>P2</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>P3</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P5</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>P6</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>P7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Training Set Pages**

- Class C1 = \{P1, P3\}
- Class C2 = \{P2, P4\}
- Class C3 = \{P5, P6, P7\}

**M(i,j)** is the value of the Feature j for the Web Page i.

The features with all true (or false) values are deleted: they surely are not useful to discriminate Classes.
• Concept Analysis is adopted to find features that characterise classes of equivalent pages

• A Concept Lattice may be automatically built on the basis of the Evaluation Matrix values (evaluated on the Training Set)
Feature Classification

According to Eisenbarth, Koschke and Simon*, features are classified, with respect to classes as:

- **Specific**, if it is satisfied by all and only the pages of the class C;
- **Relevant**: if it is satisfied by all the pages of the class C, but also by other pages from other classes;
- **CSPC** (i.e., Conditionally Specific): if it is satisfied by a subset of pages of the class C and by no other page from other classes;
- **Shared**: if it is satisfied by a subset of pages of the class C and by other pages from other classes too;
- **Irrelevant**: if it is not satisfied by any page of the classes C.

For the class **Successful Login**

- Logout is Specific
- Not(Incorrect) is Relevant
- AdminLogged is CSPC
- Not(AdminLogged) is Shared
- Login is Irrelevant
Feature Classification Evaluation

The Classification of the features may be evaluated on the basis of the Concept Analysis results:

\[
\text{Specific} (C) = \bigcap_{p \in C} \text{Intent} (OC(p)) - \bigcup_{p \in TS-C} \text{Intent} (OC(p)) \\
\text{Relevant} (C) = \bigcap_{p \in C} \text{Intent} (OC(p)) - \text{Specific} (C) \\
\text{CSPC} (C) = \bigcup_{p \in C} \text{Intent} (OC(p)) - \bigcup_{p \in TS-C} \text{Intent} (OC(p)) - \text{Specific} (C) \\
\text{Shared} (C) = \bigcup_{p \in C} \text{Intent} (OC(p)) - \text{Specific} (C) - \text{Relevant} (C) - \text{CSPC} (C) \\
\text{Irrelevant} (C) = F - \text{Specific} (C) - \text{Relevant} (C) - \text{CSPC} (C) - \text{Shared} (C)
\]
Rules for generating Discriminating Expressions

- Our goal is to obtain combinations of features (Discriminating Expressions) that are specific for a single class, i.e. that are true for any page belonging to the class and false elsewhere.

1. If the class C includes at least one specific feature, any of these specific features can be a candidate discriminating expression of the class.

2. If there are not specific features, then a candidate discriminating expression can be obtained by considering the logic and of the relevant features.
   - If there is a couple of features f1 and f2 so that f1IMP f2 (i.e. Extent(AC(f1)) ⊆ Extent(AC(f2))), then it is possible to simplify the expression by discarding the feature f2 from it.

3. If there are neither specific nor relevant features, then a candidate discriminating expression can be obtained by considering the logic or of the CSPC features.
   - If there is a couple of features f1 and f2 so that f1IMP f2, it is possible to simplify the expression by discarding the feature f1 from it.
3 - Discriminating Expression Validation

- Recall And Precision:

\[
\text{Recall} = \frac{\#(\text{pagesAttributedToClassC})}{\#(\text{AnalysedPagesBelongingToClassC})}
\]

\[
\text{Precision} = \frac{\#(\text{pagesCorrectlyAttributedToClassC})}{\#(\text{pagesAttributedToClassC})}
\]

- Failed Classification: the page class has not been identified, because no discriminating feature (or more than one) has been satisfied by the page;

- Incorrect Classification: the page has been attributed to an incorrect class;

- Correct Classification: the page has been attributed to the correct class
A Classification Example

• Web Pages related to the AdminBooks page of the open source Web application Bookstore has been considered
• Five classes of Web pages
• 25 Web pages training set
• 65 Web pages test set
A Classification Example
### Candidate features

**-First iteration-**

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>XPath Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Search Form (absolute)</td>
<td>/html/body/table[1]/tbody/tr[1]/td[1]/form[1]</td>
</tr>
<tr>
<td>2</td>
<td>Search Form (relative)</td>
<td>//form[1]</td>
</tr>
<tr>
<td>3</td>
<td>Table with Books (absolute)</td>
<td>/html/body/table[2]/tbody/tr[1]/td[1]/table[1]/tbody/tr[1]/td[1]/a[3]/font/text()=&quot;Books&quot;</td>
</tr>
<tr>
<td>4</td>
<td>Table with Books (relative)</td>
<td>//table/text()=&quot;Books&quot;</td>
</tr>
<tr>
<td>5</td>
<td>Word &quot;Next&quot; (absolute)</td>
<td>/html/body/table[2]/tbody/tr[1]/td[1]/table[1]/tbody/tr[2]/td[1]/font/a[3]/font/text()=&quot;Next&quot;</td>
</tr>
<tr>
<td>6</td>
<td>Word &quot;Next&quot; (relative)</td>
<td>//text()=&quot;Next&quot;</td>
</tr>
<tr>
<td>7</td>
<td>Word &quot;Previous&quot; (absolute)</td>
<td>/html/body/table[2]/tbody/tr[1]/td[1]/table[1]/tbody/tr[2]/td[1]/font/a[2]/font/text()=&quot;Previous&quot;</td>
</tr>
<tr>
<td>8</td>
<td>Word &quot;Previous&quot; (relative)</td>
<td>//text()=&quot;Previous&quot;</td>
</tr>
</tbody>
</table>

- 8 candidate features have been considered
- f1, f2, f3, f4 have been discarded (always true)
- Negated features have been introduced

### Classification of Candidate features

<table>
<thead>
<tr>
<th></th>
<th>Specific</th>
<th>Central Page</th>
<th>Last Page</th>
<th>Single page</th>
<th>Empty List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant</td>
<td>f5, f6,</td>
<td>f5, f6,</td>
<td>f5, f6,</td>
<td>f5, f6,</td>
<td>f5, f6,</td>
</tr>
<tr>
<td></td>
<td>f7, f8</td>
<td>f7, f8</td>
<td>f7, f8</td>
<td>f7, f8</td>
<td>f7, f8</td>
</tr>
<tr>
<td>CSPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Irrelevant</td>
<td>f5, f6,</td>
<td>f5, f6,</td>
<td>f5, f6,</td>
<td>f5, f6,</td>
<td>f5, f6,</td>
</tr>
<tr>
<td></td>
<td>f7, f8</td>
<td>f7, f8</td>
<td>f7, f8</td>
<td>f7, f8</td>
<td>f7, f8</td>
</tr>
</tbody>
</table>

**Web Pages Classification using Concept Analysis - ICSM 2007 - Paris - 4/10/2007**
Discriminating Expressions – First iteration

<table>
<thead>
<tr>
<th>Class</th>
<th>Discriminating Expression</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Page</td>
<td>f7</td>
<td>5/5</td>
<td>5/10</td>
</tr>
<tr>
<td>Central Page</td>
<td>f7</td>
<td>5/5</td>
<td>5/10</td>
</tr>
<tr>
<td>Last Page</td>
<td>f7 AND f8</td>
<td>5/5</td>
<td>5/5</td>
</tr>
<tr>
<td>Single Page</td>
<td>f7</td>
<td>5/5</td>
<td>5/10</td>
</tr>
<tr>
<td>Empty List</td>
<td>f7</td>
<td>5/5</td>
<td>5/10</td>
</tr>
</tbody>
</table>

**Candidate Discriminating Expressions**

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f3</td>
<td>Link (absolute) &quot;Next&quot;</td>
</tr>
<tr>
<td>f10</td>
<td>Link (relative) &quot;Next&quot;</td>
</tr>
<tr>
<td>f11</td>
<td>Link (absolute) &quot;Previous&quot;</td>
</tr>
<tr>
<td>f12</td>
<td>Link (relative) &quot;Previous&quot;</td>
</tr>
<tr>
<td>f13</td>
<td>Text (absolute) &quot;No records&quot;</td>
</tr>
<tr>
<td>f14</td>
<td>Text (relative) &quot;No records&quot;</td>
</tr>
</tbody>
</table>

Training Set Validation

There is not enough precision!

A new iteration of feature candidature is needed

New features have been added:
1. We need to distinguish when the words “Next” and “Previous” are links or simple labels
2. We need to distinguish between empty page and single page
Training Set Validation is OK but there are some problems in a page belonging to the Test Set (assigned to Central page instead of Last Page)

The incorrect-assigned page has been added to the Training Set and the Candidate Discriminating Expressions have been evaluated

Recall and Precision are, now, 100% both on the Training Set and on the Test Set
Discussion

- **Process Effort**
  - In the case study, about 100 minutes were needed to obtain the classification model.
  - The activities requiring human intervention were the most critical:
    - Training and Test Set Collection step
      - The reliability of the classification model depends on the *representativeness* of the Training and Test Sets.
      - Web pages resulting from the execution of a Test Suite may be good candidates to fill in the Training and the Test Set.
    - Feature Candidature step
      - Heuristic techniques allowing a semi-automatic generation of features are currently under experimentation.
Further application scenarios

- **Support to testing automation**
  - The possible reached pages are arranged in a set of equivalence classes, corresponding to different test results.
  - The proposed discriminating expressions make it possible the automatic evaluation of the testing results.

- **Support to dynamic analysis**
  - The recognition of the reached interaction scenarios makes it possible the automatic collection of logs of the visited use cases and scenarios, providing also useful information for user profiling.
Conclusions and Future Works

- A process for the definition of a classification model allowing the automatic identification of classes of Built Client pages has been defined and validated.
- The classification model allows the run-time identification of the classification of a Built Client page, with short response time and high precision.

Future Works:
- We are working for the process improvement, in order to reduce the effort needed to human made tasks.
- Further experimentation will be carried out in order to assess the scalability of the approach.
Time is over ... Are there any questions?