Semi-automated Test Planning for e-ID Systems by Using Requirements Clustering

Barış Güldalı, Holger Funke, Michael Jahnich, Stefan Sauer, Gregor Engel
18. Nov 2009
Software Quality Lab (s-lab)

- 5 software engineering professors at University of Paderborn
- 8 associated partners, 6 project partners
- 3 senior-researchers, 16 researchers

Our expertise
- Test management, test automation
- Formal methods
- Domain specific languages

Domains
- Automotive systems
- Business information systems
- Smart card systems

http://s-lab.upb.de
• The Smart Card Architects
  – Planning
  – Procuring
  – Approving

• The Core Competence
  – HJP is a team of 20 consultants with experience in all areas of eID Systems
  – Working for governments/industry in planning of e-ID and health projects in Germany, U.K., Austria, France, U.A.E., Saudi Arabia, Sudan, Korea and Japan
  – Co-editor of the ICAO Test Standards for e-passports and inspection systems
  – GlobalTester, a leading conformity test tool for e-passports and inspection systems
What is this presentation about?

- Acceptance testing of e-ID systems
- Multi-viewpoint requirements engineering
- Optimization of test plans using clustering
What is the challenge?

- Component test
- Integration test
- System test
- Acceptance test
- Conformance test

Planning
- International requirements
- National requirements
- System architecture

Test & Acceptance
- Acceptance test
- System test
- Conformance test

Customer
- Political will

Supplier
- International requirements
- National requirements
- System architecture

Consultant

Citizen

Deployment

Black Box
- Solution design
- Integration test
- Component design
- Component test
- Realization

Supplier interface

Development by supplier
What is the challenge?
What is the concrete problem?

- Reference Model for Open Distributed Processing [ISO]
- Desired overlappings in the requirements
  - Redundant test cases
  - Repeating testing steps
  - Inefficient testing

- Solution idea:
  1. Identify overlappings (similarities)
  2. Choose representatives, ignore others at first
  3. Compute an efficient order of requirements to be tested
  4. Create and execute test cases → not in the scope of this paper
### An example

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYC135M</td>
<td>The authorizing officer SHALL sign the printed application and notify the system manually that the application is authorized.</td>
</tr>
<tr>
<td>SYC480M</td>
<td>Personalization SHALL personalize the e-passport chip electrically.</td>
</tr>
<tr>
<td>SYC545M</td>
<td>An old e-passport of the recipient that is still valid MUST be cancelled.</td>
</tr>
<tr>
<td>SYE310M</td>
<td>An authorizing officer SHALL authorize the application after all necessary information has been provided (e.g., all biometric data and no violation of business rules or blacklisted).</td>
</tr>
<tr>
<td>SYE330M</td>
<td>The e-passport issue system SHALL automatically generate a production order for each authorized application.</td>
</tr>
<tr>
<td>SYE390M</td>
<td>The blank e-passport SHALL be personalized electrically, that means the signed data is stored into the contactless chip of the e-passport in a secure procedure.</td>
</tr>
<tr>
<td>SYE610M</td>
<td>After application, the current applicant's passport or e-passport SHALL be cancelled.</td>
</tr>
<tr>
<td>SYI250M</td>
<td>After authorization the e-passport issue system SHALL generate a production order from the passport application and the identity information.</td>
</tr>
</tbody>
</table>
What is this presentation about?

- Acceptance testing of e-ID systems
- Multi-viewpoint requirements engineering
- Optimization of test plans using clustering
Requirements clustering

• Related work
  ➢ functional decomposition [Hsia88, Yaung92, Otaiby05, Madhavji07]
  ➢ remodularization [Wiggerts97]
  ➢ incremental delivery [Hsia92, Hsia96]
  ➢ feature modeling [Chen05]
  ➢ requirements reuse [Lopez02]

• Testing not handled yet extensively!
  ➢ Scenario analysis [Hsia97]
  ➢ Abstraction for impact analysis [Goldin]
Questions of clustering

• What are the artifacts to be clustered?

• How is similarity between two artifacts measured?

• Which algorithm should be used for clustering of artifacts?

Informal textual requirement descriptions
- Distance measures
- Association coefficients
- Probabilistic similarity

 Hierarchical algorithms
• Partitional algorithms
Process overview

Annotation

Annotation correct?

Yes

Clustering

Clusters useful?

No (Improve Clustering)

Yes

Test plan Specification

No (Improve TP Specification) No (Improve Clustering)

Desired reduction reached?

Yes
An old e-passport of the recipient that is still valid SHALL be cancelled. Delivery SHALL notify the passport life-cycle management about the cancellation.

An old e-passport of the recipient that is still valid SHALL be cancelled. Delivery SHALL notify the passport life-cycle management about the cancellation.

(Id Actor Process Direct Object)

SYC545M-2 Delivery notify the passport life-cycle management

Linguistic analysis
Annotation – Controlled language

[Diagram with nodes labeled: CONDITIONS, ACTOR, SHOULD, MAY, PROVIDE <whom?> THE ABILITY TO <process>, BE ABLE TO <process>, OBJECT. The diagram illustrates the relationships using controlled language terms: SHALL, SHOULD, MAY, PROVIDE, BE ABLE TO.]

[Text: Obligingness [Rup07]]
Clustering

<table>
<thead>
<tr>
<th>Action</th>
<th>State</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>apply</td>
<td>send</td>
<td>have support</td>
</tr>
<tr>
<td>personalize</td>
<td>receive</td>
<td>be conform to</td>
</tr>
<tr>
<td>deliver</td>
<td>update</td>
<td>comprise allow</td>
</tr>
</tbody>
</table>

Partitional Clustering

### Similarity function

\[
s(r_1, r_2) = \begin{cases} 
  3 & \text{if Actor, Process and direct Object are same} \\
  2 & \text{if Actor and Process are same} \\
  1 & \text{if Actor is the same} \\
  0 & \text{otherwise} 
\end{cases}
\]

<table>
<thead>
<tr>
<th>Id</th>
<th>Actor</th>
<th>Process</th>
<th>Direct Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYC535M</td>
<td>The recipient</td>
<td>provide</td>
<td></td>
</tr>
<tr>
<td>SYE430M-2</td>
<td>The recipient</td>
<td>provide</td>
<td>his ID card and the receipt</td>
</tr>
<tr>
<td>SYE430M-3</td>
<td>the recipient</td>
<td>provide</td>
<td>his current e-passport</td>
</tr>
<tr>
<td>SYE630M</td>
<td>the recipient</td>
<td>provide</td>
<td>his receipt and his ID card</td>
</tr>
<tr>
<td>SYE640M</td>
<td>the recipient</td>
<td>provide</td>
<td>his receipt and his ID card</td>
</tr>
</tbody>
</table>
Test plan specification

- **Heuristic**
  - Test activity: Requirement to be tested
  - Test step: Action1
  - Test step: Action2
  - Test step: ActionX (r2)
  - Test step: Action2 (r1)

- **Pattern 1**
  - Action/State/Rule
  - Subject
  - verb
  - Object

- **Pattern 2**
  - Action
  - Subject
  - verb
  - Object
  - State/Rule
  - r1
  - r2
## Tool support

<table>
<thead>
<tr>
<th>Activity</th>
<th>Degree of automation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation</td>
<td>high</td>
</tr>
<tr>
<td>• Splitting</td>
<td></td>
</tr>
<tr>
<td>• Syntactical analysis</td>
<td></td>
</tr>
<tr>
<td>Clustering</td>
<td>high</td>
</tr>
<tr>
<td>• Coarse grained</td>
<td></td>
</tr>
<tr>
<td>• Fine grained</td>
<td></td>
</tr>
<tr>
<td>Test plan specification</td>
<td>medium</td>
</tr>
<tr>
<td>• Test plans</td>
<td></td>
</tr>
</tbody>
</table>

Prototype implemented
Case study

Statistics on analyzed requirements

<table>
<thead>
<tr>
<th>Viewpoints</th>
<th># of Req.</th>
<th>Action</th>
<th>State</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise</td>
<td>103</td>
<td>84</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>Information</td>
<td>61</td>
<td>25</td>
<td>29</td>
<td>7</td>
</tr>
<tr>
<td>Computation</td>
<td>141</td>
<td>89</td>
<td>39</td>
<td>13</td>
</tr>
<tr>
<td>Engineering</td>
<td>67</td>
<td>27</td>
<td>26</td>
<td>14</td>
</tr>
<tr>
<td>Technology</td>
<td>33</td>
<td>0</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>405</strong></td>
<td><strong>300</strong></td>
<td><strong>72</strong></td>
<td><strong>11</strong></td>
</tr>
</tbody>
</table>

Number of requirements in clusters

<table>
<thead>
<tr>
<th>Viewpoints</th>
<th>Enrolment</th>
<th>Personalization</th>
<th>Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial</td>
<td>final</td>
<td>initial</td>
</tr>
<tr>
<td>Enterprise</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Information</td>
<td>0</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>Computation</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Engineering</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Technology</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>13</strong></td>
<td><strong>34</strong></td>
<td><strong>13</strong></td>
</tr>
</tbody>
</table>

Performance of steps in milliseconds

<table>
<thead>
<tr>
<th># of Req.</th>
<th>Annotation</th>
<th>Similarity</th>
<th>Clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4.594</td>
<td>&lt;1</td>
<td>103</td>
</tr>
<tr>
<td>50</td>
<td>19.938</td>
<td>9</td>
<td>1.534</td>
</tr>
<tr>
<td>100</td>
<td>35.797</td>
<td>15</td>
<td>4.602</td>
</tr>
<tr>
<td>400</td>
<td>141.859</td>
<td>31</td>
<td>55.198</td>
</tr>
</tbody>
</table>

Detected similarities: 73
Threshold: 2

After applying heuristics
After applying patterns
Conclusion

• Overlappings and timing relations can be detected
  – Linguistic analysis
  – Clustering

• Important: What is the target?

• Efficient test planning possible
  – Eliminating overlappings in test plans
  – Ordering related requirements in test plans

• Tools for particular tasks available

• Lessons learned for specifying requirements
  – No passive voice
  – Use atomic sentences
  – …
Outlook

• Test case generation from test plans (Summer 2010)
  – Choosing testing technique → Interaction rules, e.g. human-to-system interaction
  – Concrete test data → e.g. boundary analysis, equivalence classes

• Current master‘s thesis (Spring 2010)
  – Similarity function refined
  – Classification for general verbs improved [Dowty79]
  – Further improvements on prototype, e.g. statistics, warning

• Automation of test execution
  – Testing GUI, Reporting

• Problems and solutions apply also to other domains
Thank you for your attention!

Questions?

Barış Güldalı
Researcher

Software Quality Lab (s-lab)
Universität Paderborn
Warburger Str. 100
33098 Paderborn/Germany
+49 5251 602416

http://s-lab.upb.de
bguldali@s-lab.upb.de
## Literature on requirements clustering

<table>
<thead>
<tr>
<th>#</th>
<th>Abr.</th>
<th>Authors</th>
<th>Titel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clustering algorithms in general</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Jai02</td>
<td>Jain, Murty, Flynn</td>
<td>Data Clustering: A Review</td>
</tr>
<tr>
<td>2</td>
<td>Pro97</td>
<td>Procopiuc</td>
<td>Applications of Clustering Problems</td>
</tr>
<tr>
<td>3</td>
<td>Wig97</td>
<td>Wiggerts</td>
<td>Using Clustering Algorithms in Legacy Systems Remodularization</td>
</tr>
<tr>
<td>4</td>
<td>Voo86</td>
<td>Voorhees</td>
<td>Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval</td>
</tr>
<tr>
<td>5</td>
<td>Def78</td>
<td>Defays</td>
<td>An efficient algorithm for a complete link method</td>
</tr>
<tr>
<td><strong>Requirements Clustering Algorithms</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mad07</td>
<td>Li, Rahman, Madhavji</td>
<td>An Approach to Requirements Encapsulation with Clustering</td>
</tr>
<tr>
<td>7</td>
<td>Che05</td>
<td>Chen, Zhang, Zhao, Mei</td>
<td>An Approach to Clustering Feature Models Based on Requirements Clustering</td>
</tr>
<tr>
<td>8</td>
<td>Ota05</td>
<td>Otaiby, AlSherif, Bond</td>
<td>Towards Software Requirements Modularization using Hierarchical Clustering Techniques</td>
</tr>
<tr>
<td>9</td>
<td>Lop02</td>
<td>Lopez, Laguna, Garcia</td>
<td>Reuse based Analysis and Clustering of Requirements Diagrams</td>
</tr>
<tr>
<td>10</td>
<td>Hsia96</td>
<td>Hsia, Hsu, Kung, Holder</td>
<td>User-Centered Systems Decomposition: Z-based requirements Clustering</td>
</tr>
<tr>
<td>11</td>
<td>Yau92</td>
<td>Yaung</td>
<td>Design and Implementation of a Requirements Clustering Analyzer for Software System Decomposition</td>
</tr>
<tr>
<td>12</td>
<td>Hsi92</td>
<td>Hsia, Gupta</td>
<td>Incremental Delivery Using Abstract Data Types and Requirements Clustering</td>
</tr>
<tr>
<td>13</td>
<td>Hsi88</td>
<td>Hsia, Yaung</td>
<td>Another Approach to System Decomposition: Requirements Clustering</td>
</tr>
<tr>
<td><strong>Re Tools</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>iX07</td>
<td>Hood, Mühlbauer, Rupp, Versteegen</td>
<td>IX Studie: Anforderungsmanagement – Methoden und Techniken, Einführungsszenarien und Werkzeuge im Vergleich</td>
</tr>
<tr>
<td>15</td>
<td>Gei07</td>
<td>Geisser, Hildenbrand, Riegel</td>
<td>Evaluating the Applicability of Requirements Engineering Tools for Distributed Software Development</td>
</tr>
<tr>
<td>16</td>
<td>Kra06</td>
<td>Krauß, Versteegen, Mühlbauer</td>
<td>Model Driven Requirements Engineering</td>
</tr>
<tr>
<td>17</td>
<td>Ver05</td>
<td>Versteegen</td>
<td>Anforderungsmanagement-Werkzeuge im Vergleich</td>
</tr>
</tbody>
</table>
Test plan meta-model
Partitional clustering

1. Setup similarity matrix
2. Initialize clusters
3. Merge
4. Partition
5. Check halting criteria
6. Closure