Interactive Visualization of Modifications in Software Models

Jakub Ondík
supervisor: Dr. Karol Rástočný
Slovak University of Technology

Motivation
- Evolution of software models
- Provide information about modifications for both software analysts and developers
- Traceability of customer's requests
- Provide basis for source code modifications which need to be propagated from models
- Support for team discussion

Visualization of Modifications
- Existing solutions
  - Table View
  - Side by side comparison
- Our method
  - Single diagram visualization
  - Lesser cognitive load

Evaluation
- 9 participants, 5 diagrams
- First day - A and B versions side by side
- Second day - our method

![Diagram comparison](image)

<table>
<thead>
<tr>
<th>Diagram</th>
<th>Side-by-side [s]</th>
<th>Our method [s]</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>19</td>
<td>10</td>
<td>43.36%</td>
</tr>
<tr>
<td>D2</td>
<td>24</td>
<td>11</td>
<td>54.16%</td>
</tr>
<tr>
<td>D3</td>
<td>55</td>
<td>12</td>
<td>78.18%</td>
</tr>
<tr>
<td>D4</td>
<td>28</td>
<td>9</td>
<td>67.66%</td>
</tr>
<tr>
<td>D5</td>
<td>75</td>
<td>14</td>
<td>81.33%</td>
</tr>
</tbody>
</table>

Conclusions
- Our method provides faster identification of modifications in software models
- As a basis for visualization, we have created a modifications detection method for UML
- Future work can consist of adding wider range of supported diagram types and evaluation in an industrial environment