Implementation of Collaborative Data and Schema Conflict Resolution in EA Repositories
Guided Research: Introductory Presentation; 8.7.2013

Referee: Björn Kirschner
Supervisor: Sascha Roth

Software Engineering betrieblicher Informationssysteme (sebis)
Ernst Denert-Stiftungslehrstuhl

www.matthes.in.tum.de
Agenda

1. Motivation
2. Research Questions
3. Proposed Meta-Information Model
4. Related Work
5. Timeline
6. Conclusion
Motivation: Co-Evolution of Models in an EA
How can model conflicts in hybrid wiki workspaces be solved collaboratively?

- How does a meta-information model, which supports the collaborative solving, look like?

- How would an algorithm for merging co-evolving EA models look like?

- How can users be involved for the collaborative solving process?
Meta-Information Model in Hybrid Wikis

Model

Object

Definition

Source: [1]
Types of tasks:
- Conflict
- Validate
- Approve
### Related Work: Collaborative Handling of Conflicts

<table>
<thead>
<tr>
<th>Insert</th>
<th>Delete</th>
<th>Update</th>
<th>Use</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td>Insert</td>
</tr>
<tr>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td>Delete</td>
</tr>
<tr>
<td>×</td>
<td></td>
<td>~</td>
<td>~</td>
<td>Update</td>
</tr>
<tr>
<td>~</td>
<td></td>
<td></td>
<td>~</td>
<td>Use</td>
</tr>
<tr>
<td>×</td>
<td></td>
<td></td>
<td>~</td>
<td>Move</td>
</tr>
</tbody>
</table>

- × ... *Conflict*
- ~ ... *Warning*

Source: [2]
Timeline

- **Conceptual design**
- **First steps in Tricia: Roles**
- **TDD: Implementation of the merging algorithm**
- **Writing the paper**
- **Submission**

- **Today**
- **September**
- **November**
Prospects

- Publication of a conference paper
- Technical evaluation via a prototype implementation in Tricia
- Tobias‘ thesis: Task handling via visualizations
- For a future master‘s thesis: evaluation of the concept in the industry

Implementation: Complications

Tricia Object Model:
Algorithm 1: n-way merging of models

Data: Model $M_{1..n}$, Baseline Time $t_b$, Target Model $M_t$
Result: Task $T_{1..n}$

1: conflicts ← \{(key: $\emptyset$, value$^b$: $\emptyset$, value: $\{\emptyset\}$\}
2: approve ← \{(key: $\emptyset$, value$^b$: $\emptyset$, value: $\{\emptyset\}$\}
3: validate ← \{(key: $\emptyset$, value$^b$: $\emptyset$, value: $\{\emptyset\}$\}
4: changesets ← $\{\emptyset\}$
5: // 1. collect operations
6: foreach element $\in M_{1..n}$ do
7:     foreach changeset $\in element$ do
8:         if changeset.when $> t_b$ then
9:             changesets ← changesets $\cup$ changeset
10: // 2. consolidate operations
11: foreach $c_1 \in$ changesets do
12:     foreach $c_2 \in$ changesets do
13:         conf = detectConflicts($c_1$, $c_2$)
14:         // see Algorithm 2 and 3
15:         if conf $\neq \emptyset$ then
16:             if $\exists c_{1.modelElement} \in$ conflicts then
17:                 chgConf ←
18:                     conflicts.getAndRemove($c_{1.modelElement}$) $\cup$
19:                     conf
20:             conflicts ← conflicts
21:                 $\cup(c_{1.modelElement}, c_{1.modelElement}^b, chgConf)$
22:         else
23:             conflicts ← conflicts $\cup$ ($c.modelElement, c$)
24:         // 2. apply to target model
25:         foreach $c \in$ changesets do
26:             if $e \equiv c.modelElement$ then
27:                 changesets.remove($c$)
28:     // 4. check consistency/schema conformance
29:     // 5. user intervention
30:     // 6. create conflict resolution tasks
31:     // notify role of e role of change unify them in a group, notify
32:     group

Björn Kirschner - Implementation of collaborative data and schema conflict resolution in EA repositories – 8.7.2013
© sebis 12