CIB UNTERNEHMENSGRUPPE

GSOC
SAVING ODF XML OF CHANGE-TRACKING AS A SEQUENCE OF PRE-DEFINED CHANGES
Documents alone are not sufficient for interoperable collaboration
“If I had asked people what they wanted, they would have said faster horses”, Henry Ford
Document collaboration evolved greatly from floppy discs over Email to Cloud Space...

...but only as long there is only one editing at the time..

There can only be one.. editor
Faster Floppy Discs, Email, Cloud...
Main Collaboration Problem: Merging

- **Use Case 1:**
  Problems after returning documents from business partner / business software
  - What has been changes?
  - All changes guaranteed listed via change-tracking?
  - How merging multiple document versions?

- **Use Case 2:**
  Real-Time editing among different ODF applications is impossible
  - Today only complete ODF Documents can defined to be exchanged!
  - We need to send a single ODF change without document
MERGE REQUIREMENTS

- **FIND DIFFS:**
  Finding differences between two ODT documents

- **CHANGE STEPS:**
  Change sequences to apply each other changes

- **INTEROPERABILITY:**
  ODF ecosystem blossom with interoperable way to serialization changes (e.g. as XML & JSON)
**ODF CHANGE DESIGN**

- High Level Abstraction from ODF XML -> avoids diffing noise
- Logical / Semantic User Changes -> best applicable to different models
- Atomic Change -> allows switching two changes without context

<table>
<thead>
<tr>
<th>ODT Document</th>
<th>ODT Changes (4 for HEADING only in JSON)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HEADING</strong></td>
<td>{&quot;name&quot;:&quot;addPara&quot;,&quot;start&quot;:[1],</td>
</tr>
<tr>
<td></td>
<td>&quot;attrs&quot;:{&quot;style&quot;:&quot;Heading_20_1&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;paragraph&quot;:{&quot;outlineLevel&quot;:1}</td>
</tr>
<tr>
<td>1. item1</td>
<td>}</td>
</tr>
<tr>
<td>2. item2</td>
<td>}</td>
</tr>
</tbody>
</table>

{"name":"addText","start":[1,1],"text":"HE"},
{"name":"addText","start":[1,3],"text":"A"},
{"name":"addText","start":[1,4],"text":"DING"},

---
How to solve design decision problems?

Rule of thumb:
When in doubt, go for the design which offers the greater feature set (customer first)

Example:
Ids vs. Number references
Remote reference on read-only document possible only by number
COLLABORATION BASED ON CHANGES

Information System

ODT Document

ODF Sequencer

ODF Web Editor using changes

ODT Changes

ODT Changes

ODF Web Editor using changes

ODT Changes
NEXT STEPS

- ODF Sequencer (Java, ALv2)
- Upcoming Apache ODF Toolkit (Incubator) feature
- ODT to „JSON Changes“
- Applying new changes to ODT
- LibreOffice automatic tests for ODT filter (feature level diff)
  - Tool to filter single feature bug, so not all testing is blocked
- Prototype ODF Git integration
  - ODF changes as diff
  - ODF changes to merge
  - Using ODF Sequencer
NEXT STEPS

- Work on OASIS ODF Changes specification
- LibreOffice Change-Tracking based on „OASIS ODF changes“
- Improve LibreOffice Tracking (Format on Text / Table / Styles Tracking)
Support ODF!
LibreOffice strongest ODF player benefits most!!
ISO standard has enormous marketing effects!!!
Leverage of ISO standards on MS ecosystem:
  Name of ODF alpha ver. = Open Office XML
  Name of MS doc format = Office Open XML
Interoperable Collaboration
  a silver bullet for future business software?

REMEMBER
GROUP UP!
PLAY SMART!