Automated Format Transformation for Courseware
by
Lutz Finsterle and Martin Rotard

Finsterle@ikr.uni-stuttgart.de
Rotard@informatik.uni-stuttgart.de

Agenda

- About the Speakers
- Requirements in the E-Learning Project ITO
- Transforming OpenOffice documents into the ITO intermediate courseware format
- Converting Slide Presentations to SVG
- Open Issues and Ideas
- Conclusion and Outlook
About the speaker

Lutz Finsterle
- Researcher and staff-member of the
  Institute of Communication Networks and
  Computer Engineering (IKR)
  University of Stuttgart
- Engaged in the multimedia teaching projects CANDLE (EU) and ITO (BMBF)

Martin Rotard
- Researcher and staff-member of the
  Visualization and Interactive Systems
  Group (VIS), University of Stuttgart
- Engaged in the multimedia teaching project ITO (BMBF)

The E-Learning project ITO

- "Information Technology Online"
- Funded by the German Federal Ministry of
  Education and Research" (BMBF)
- Goal: Exchange materials for lectures
- Associated universities:
  Uni Stuttgart, TU München, TU Hamburg-Harburg, TU Dresden, PH Ludwigsburg
### Requirements in the ITO-Project

- Keep the well-known authoring tools (OpenOffice, Microsoft Word, Adobe FrameMaker, LaTeX, etc.)
- But exchange and reuse of newly generated and existing content
- Solution: Definition of an intermediate courseware format in XML
- Developed Tools:
  - Transforming OpenOffice documents into the ITO intermediate courseware format
  - Converting slide presentations to SVG
  - Distributed web-portal for courseware

### Structuring the Material

![Diagram showing the process of structuring material](image)

- MS Word *.doc
- MS PowerPoint
- OpenOffice *.sx[dl],*.rtf
- FrameMaker
- TeachML
- TeX/ LaTeX
- OpenOffice
- OOfficeML
- Transformer
- Generic XML
- ITOML
  - Content
  - Metadata
  - Objects
  - Structure
  - Documents
  - Layout
Delivery of Content

Prerequisites to Transformation

- Wide variety of authoring tools supported
- Wide variety of formats supported
- XML based format to start from
- Means to Identify certain passages
  - During authoring
  - When transforming
- Separation of content, layout, ...
- Definition of output format
- Two-way transformation (should) be possible
Transforming OO documents into the ITO intermediate courseware format

- Definition of style templates
- Paragraph styles
  - introduction, motivation, definition, rights, example, conclusion etc.
- Character styles
  - keyword, index, reference, citation, significant, strong significant etc.
Overview on ITOML

Hierarchical Course Structure
Steps in Transformation

- Starting from source format
- Converting to OpenOffice format (External Java application at the moment)
- Open content.xml
- Identify used styles
- Translate auto-styles
- Walking through document and build the new one
- Store in whole or as sub-parts

Conclusions

- Using style-formats is well-known
- Still, instruction of authors is needed
- To help the authors, a FAQ has been build up
- Verifier is at the moment being implemented, that helps the author to find structural errors
- Transformation-Engine in rewrite to more complex structure
- First test have shows very promising results
Converting slide presentations to SVG

- Slide presentations are most commonly used media to present lecture material today.
- Huge amount of time has been invested to generate.
- Preserve this when assembling online courseware.
- Seamless transformation to the web-enabled format SVG.
- SVG: upcoming W3C standard for two-dimensional graphics on the web.
- OpenOffice allows the export of slides into SVG, unfortunately only one slide at a time.

Converting slide presentations to SVG

optional

Impress
PowerPoint ...

import
controls

export

controls
(UNO interface)

Add navigation, SVG-optimization, etc.

OpenOffice.org
The Open Source Office Suite

SVG

JAVA

2003-03-20
Converting slide presentations to SVG (2)

Features
- Export of the complete presentation
- Adding navigation and index mechanisms
- Correction of glyph spacing issues
- On-the-fly embedding of non-standard fonts

Planned/Partially Implemented Features
- Support for animations
- Support for annotations

On the availability of Fonts

Starting Point
- Font declarations in exported SVG such as font-family, font-style, ...
- Build SVG-Fonts from true-type fonts
- Embed fonts into the exported SVG

Problems
- Copyright issues
- Full description of SVG-Fonts produced form ttf must be added by hand
- Embedded SVG-fonts do not work on all viewers

Solutions
- Put only the needed glyph declarations into the final document
- If not importing a non standard font, automatic reduction of <tspan>-coordinates information
Adapting Animations

- **Starting Point**
  - Animation information in OpenOffice native format
  - Animated objects can be identified in native format
  - Animations must be modelled in SVG

- **Problems**
  - Animation information do not make their way to the exported SVG
  - Even 1:1 object identification not easily possible
  - Modelling animation in SVG is tedious cause of missing features in SVG-viewers

- **Solutions**
  - Animations are realized in Java-Script
  - Mapping is done by hand right now
  - Extend SVG export engine to transport information to exported SVG

Technical and Other Problems

- **Embedding of Fonts**
  - Copyright problems with Fonts
  - Font selection in SVG Viewers
  - SVG-embedded-font abilities of different SVG-Viewers

- **Animations**
  - Correlation of animations to SVG object
  - Scripting abilities of viewer
  - SVG-SMIL implementations not fully working
### Links & questions

- **IKR**  
  [http://www.ikr.uni-stuttgart.de](http://www.ikr.uni-stuttgart.de)

- **VIS**  
  [http://wwwvis.informatik.uni-stuttgart.de](http://wwwvis.informatik.uni-stuttgart.de)

- **ITO Project**  
  [http://www.ias.uni-stuttgart.de/ito/](http://www.ias.uni-stuttgart.de/ito/)

- **CANDLE**  
  [http://www.candle.eu.org](http://www.candle.eu.org)

- **OpenOffice.org**  
  [http://www.openoffice.org](http://www.openoffice.org)

### Questions?