Template Based Authoring for AR based Service Scenarios

Christian Knöpfle, Fraunhofer IGD
Jens Weidenhausen, Fraunhofer IGD
Laurent Chauvigné, BMW AG
Ingo Stock, BMW AG

email: knoepfle@igd.fhg.de
Outline

Motivation

State of the Art

A short look into reality

Template Based Authoring

Demo!

Conclusion
Motivation

The well known challenge of product development:

Increase of complexity, but decrease of development time

Speed up product development by using Virtual Reality!

But the process doesn’t end with development!

• Production
• Service and Maintenance
Motivation

Service and Maintenance:

• Training takes time
• Writing manuals takes time
  - Photos of real car
  - Translation in other languages
• Using the manuals is complicated

Installation:
Replace sealing ring on spacer.
Install servomotor (1) and screw in up to spacer.
Rotate servomotor (1) into correct installation position.
Install and tighten down screws.
Motivation

Trend towards Augmented Reality, because

- context sensitive information presentation
  - Reduce training
  - Reduce errors
- Graphical (universal language)
- Mobile
Motivation

Sounds good, but where is the data coming from?

- Modeling in 3DS Max
- Position models relative to markers
- Writing Javascript code
- Integrate in AR-System

Drawbacks

- It takes a lot of time
- You need to be a programmer
- Technical writers are not programmers!

Top priority: It must be a tool for technical writers!
Some existing solutions

AMIRE (Zauner et. al., ISMAR2003)
- Markers are objects and can trigger actions
- Fields and Routes concept
- It is a good programmers tool…

PowerSpace (Haringer, Regenbrecht, ISMAR02)
- Powerpoint based scene description, then rearrange in PowerSpace
- Interactive placement of annotations
- No linkage between annotation and scene objects
A short look into reality

Real world scenarios (Focus on automotive)

- Change oil filter
- Replace driving belt
- Valvetronic motor replacement

Similarities:

- Consist of atomic work tasks (called “operations”)
  - Loosen part, remove part, screw, measure, …

- Most operations are similar they only differ in their parameters, e.g. release takes parameter *tool* and *part*

- Specific chronological order
Next, the worker should remove the screw with the ratchet.

The building blocks:

- Timeline (temporal order of action playback)
- Actions (the operation)
- Objects (parts of scene, tools)
Timeline

Basic Concept:

- Similar to video editing programs
- Each action is wrapped in a time container
- Each container has a start time and a duration
- Each container could be positioned anywhere on the timeline

[Diagram of a timeline with actions: Open hood, Remove screw, Remove screw, Remove screw, Remove screw, Remove cover]
Objects

Concept:

- Represents a “real” part or tool in the virtual scene
- Include meta data, depending on their type, e.g.
  - **Screws**: Contact point, possible movement direction
  - **Tools**: Contact point, animation of tool and resulting effect on connected part
- Implementation: VRML97 prototypes with defined API
Actions

Concept:

- Represents an operation
- Defines the graphical representation (e.g. additional arrows)
- Takes care of animations
- Meta data for parameter definition (kind of tools etc.)
- Implementation: VRML97 prototypes

Examples of actions:

- Release, Fasten
- Screw in
- Remove
Template Based Authoring

Everything put together in a GUI

Unscrew Action

The Tool

The Part

creation of Action-GUI is based on metadata
Playback of actions

- Object defines initial pos and ori (time = 0)
- Tool defines subsequent pos and ori
The Advantages of Template Based Authoring

• High level authoring (well suited for non-programmers)
• No need to think about the graphical representation
• Easy realization of style guide
• Automatic creation of text descriptions

• Output:
  • Still images for pdf-Style documentation
  • Movies for VR documentation
  • Scene description for AR manuals
Conclusion

• Flexible and extensible concept for creation of service manuals
• The basic blocks: Timeline, Object, Action
• Adapts the way technical writers think

• Move to X3D for better encoding of meta data
• Improve object interface (e.g. transparent objects)
• Object dependencies
• Ongoing work: IST-ULTRA (www.ist-ultra.org)
Thanks for your attention!

Questions?