CONTIGRA:
An XML-Based Architecture for Component-Oriented 3D Applications

Raimund Dachselt, Michael Hinz, Klaus Meißner
Dresden University of Technology, Multimedia Technology Group

Outline

- Motivation and Vision
- Related Work
- The CONTIGRA Architecture
  - Component Development Levels & Tasks
  - Three Levels & their Markup Languages
  - Example 3D Application
- Conclusion & Future Work
Motivation

Current Situation

- Improvements in 3D graphics hardware & fast-evolving (3D) Internet technologies
- Increase of Web-based 3D applications
  – E-Commerce, Entertainment, Education, InfoVis…
Motivation

😊 Problems

- Multitude of proprietary Web3d-formats | X3D
- Lack of 3D design standards, guidelines
- Authoring tools proprietary, not multi-disciplinary
- Too much programming, time-consuming | Reuse?

😊 Future Vision

- Repertoire of adaptable 3D Widgets, Metaphors
  → Standards for 3D user interfaces & desktop VE
- Reuse of high-level 3D components
Motivation / Problems

- Multitude of proprietary Web3d-formats | X3D
- Lack of 3D design standards, guidelines
- Authoring tools proprietary, not multi-disciplinary
- Too much programming, time-consuming | Reuse?

Future Vision

- Repertoire of adaptable 3D Widgets, Metaphors
- Standards for 3D user interfaces & desktop VE
- Reuse of high-level 3D components

Component

- **WidgetComponent**
  - SelectionComponent
  - ...
  - NavigationComponent
  - ...
  - ManipulationComponent
  - ...
- **ApplicationControlComponent**
  - ButtonComponent
  - ...
  - ValuatorComponent
  - ...
  - TextInputComponent
  - WindowComponent
  - **MenuComponent**
  - RingMenuComponent
  - ...
  - ToolTipComponent
  - ColorChooserComponent
  - ...
  - VisualizationComponent
  - ...
  - OrientationComponent
  - ...
- **AvatarComponent**
- **AnimationComponent**
Related Work

- Few 3D Component Approaches
  - NPSNET-V, Bamboo, i4D, 3D Beans, …
  - Jamal (BML), 3dml, VRML Prototypes
  - 3D format dependency, code-centered, low-level

- X3D - extensibility concepts
  - Prototypes, Profiles, Components
  - Basically Scene Graph Extensibility
  - No SG abstractions, no higher-level assembly
CONTIGRA Architecture

Component OrienNted Three-dimensional Interactive GRaphical Applications

Characteristics

- Document-centered 3D component architecture
- Documents describe component interfaces, implementation, configuration, and assembly
- Multi-layered, declarative approach (basis: XML)
- High-level view, hides scene graph details
- Abstraction to existing 3D toolkits, formats, APIs largely independent of implementation issues
### Component Levels & Tasks

<table>
<thead>
<tr>
<th>Level</th>
<th>Tasks</th>
<th>Result (Documents)</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime</td>
<td>Usage Adaptation</td>
<td>Executable 3D Application (Web / stand alone)</td>
<td>Specific 3D Viewer (X3D Applet, VRML-, Viewpoint PlugIn …)</td>
</tr>
<tr>
<td>Configuration &amp; Assembly</td>
<td>Configuration Assembly Linking</td>
<td>Assembled 3D Application (format independent)</td>
<td>CONTIGRA SceneBuilder</td>
</tr>
<tr>
<td>Distribution</td>
<td>Selection Retrieval</td>
<td>Packaged 3D-Components</td>
<td>Component Database, Web Interface</td>
</tr>
<tr>
<td>Development</td>
<td>Description</td>
<td>Component Interface</td>
<td>CONTIGRA ComponentBuilder</td>
</tr>
<tr>
<td></td>
<td>Implementation</td>
<td>Implementation Files (Media Standards)</td>
<td>3D-Modelling, Media &amp; Programming Tools</td>
</tr>
<tr>
<td>Task</td>
<td>XML Schema</td>
<td>CONTIGRA Documents</td>
<td>Other Resources</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------</td>
<td>--------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CONTIGRA Markup Levels**

- **CONTIGRA SceneGraph**
  - X3D, Audio3D, Behavior3D

- **Scene Graph Integration and Linking**

- **<CoSceneGraph>**
  - Audio Graph
  - Behavior Graph
  - Geometry Graph
  - SceneGraphs

- **Component**
  - Sound
  - JAR, Scripts
  - Video, Graphics

- **X3D Profiles**
Level 1: SceneGraph

- Component implementation language, integrates various scene graph & media files

- Scene graph parts referenced using XLink with XPointer syntax
**CONTIGRA Markup Levels**

<table>
<thead>
<tr>
<th>Task</th>
<th>XML Schema</th>
<th>CONTIGRA Documents</th>
<th>Other Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Assembly</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Linking</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONTIGRA SceneComponent&lt;br&gt;&lt;CoComponentInterface&gt;</td>
<td>Component Interface Declaration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONTIGRA SceneGraph&lt;br&gt;&lt;CoSceneGraph&gt;</td>
<td>Scene Graph Integration and Linking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONTIGRA Components&lt;br&gt;Audio3D, Audio3D, Behavior3D</td>
<td>Audio Graph&lt;br&gt;Behavior Graph&lt;br&gt;Geometry Graph</td>
<td>X3D Profiles</td>
<td></td>
</tr>
</tbody>
</table>

**Icons:**
- **Child Components**
- **Editors**
- **Icon**

**Additional Resources:**
- Sound
- JAR, Scripts
- Video, Graphics

**XML Schemas:**
- XML Schema
Level 2: SceneComponent

Description

Diagram showing the structure of SceneComponent with components such as Header, Info, ComponentClass, Documentation, Deployment, AuthoringSupport, Semantics, LicenceModel, Implementation, CoSceneGraphFile, csc:ParameterListType.
Level 2: SceneComponent

Interface & Configuration

- mappedParameter
- abstractParameter
- applicationParameter
- geometryPart
- audioPart
Level 2: SceneComponent

Assembly & Linking
Level 3: Scene

- High-level component integration language
- Requirements for specific 3D web environment
3D Application Example
3D Application Example

No animals were harmed during the production of this demo
Conclusion & Future Work

- Major Features
  - Componentization (design | deployment)
  - Reuse on different abstraction levels
  - Platform independence, abstraction to 3D formats
  - Declarative approach, well suited for tool support

- Next Steps
  - CONTIGRA ComponentBuilder | SceneBuilder
  - Development of runtime-framework (translators)
  - More 3D widgets and example applications
Thank you for your attention!

www.CONTIGRA.com