An Efficient Object-oriented Authoring and Presentation System for Virtual Environments

R. Dörner, W. Mueller, V. Luckas, A. Schaefer
Fraunhofer Institute for Computer Graphics
Darmstadt University of Technology
Darmstadt, Germany
doerner@igm.fhg.de
Outline

- Motivation and requirements
- System architecture
- System concepts
- System implementation
- Applications
- Conclusion
Virtual Environments

- **New Opportunities**
  - WWW-based: VRML, Java 3D
  - Distributed multi-user environments
  - Increasing performance

- **Old Problems**
  - High efforts in time and cost for authoring
  - Presumption of certain skills
Requirements

• Multi-purpose systems
  – Interactive virtual environments
  – Semi-interactive visualizations
  – Presentations
  – Non-interactive photo-realistic animations

• Different user groups
  – Professional users
  – Users without VR-specific knowledge
Innovative Authoring Concepts

- Authoring process itself has to be improved
- In graphics authoring *Clipart* is a common concept

⇒ How do reusable components for animation authoring look like?
⇒ Is it possible to have Clipart in 3D animation?
Object-Orientation

• Applied to the system (e.g. abstract rendering class)

• Applied to the authoring (e.g. entities in the virtual environment are modelled as instances of classes)
Architecture Overview

- Simulator
- SPS Control
- Control Devices
- Scene Editor
- Animation System
- Animation Element Library
- Modeler
- Behavior Editor
- Presentation Toolkit
- Renderer
- VR-Viewer
- Behavior
- Geometry
Architecture

- Several independent and coordinated tools
- Geometry and behavior
- Flexible concept already used by animation systems (e.g. Clockworks) or visualization systems (e.g. Khoros) applied to support interactive and behavioral entities
Definition: Animation Element (I)

- Analogous to Clipart
- Animation Element: Not Geometry only
Definition: Animation Element (II)

- is an independent entity
- represents an object of the user’s “world”
- comprises a description of
  - visual appearance
  - specific behavior (animation definition, application logic, interaction facilities)
Visual Design

- Identifying animation elements
- Design **not** for one specific application
- Representation forms

- Level of detail
Behavior

• Providing of functionality
• Basic functionality vs. element-specific functionality
• Example: *scale*
Functionality with Parameters

• Object-specific “intelligence”
• Example: *walk* (parameter: speed)

slow  fast
Object-specific State
Scripting Interface - Example

```
peter = new Person( );
peter.scaleLength( 1.82 );
peter.setExpression( Expression::ANGRY);
pos1 = new Position( 3, 0, 5 );
pos2 = new Position( 1, 0, 1 );
time  = new Time( 0.0 )
peter.walk( time, time+5, pos1, pos 2 );
```
Scripting Interface

• Scripting interface should be transparent for the user

• Example: Script generated automatically by mapping scene editor information and simulation events

• Compilation of the script leads to internal representation that is used to support different output formats
The CASUS System

• CASUS
  – Computer Animation of Simulation Traces
• A tool for automatic generation of 3D animation from event oriented simulator data
• Developed as part of the Demonstration Centre Simulation in Production and Logistics
Authoring

- System programers implement different linkage modules and output drivers
- Element programmers model and implement animation elements
- Authors create and edit scenes
- Users view and/or interact with the scene
CASUS Base: A Library
CASUS Anim: An Animation System
Authoring with Simulator Linkage
Example: VRML Output

- Geometry: VRML  Behavior: Java
- Using the VRML external authoring interface
CASUS Presenter

International Conference Graphicon 1998,
Moscow, Russia, http://www.graphicon.ru/
Flexibility

• Design of each tool should keep multi-purpose requirement in mind

• Example: CASUS Presenter
  – Platform independence (PC to high-end graphics workstation)
  – VR scalability (ranging from desktop VR to immersive VR, e.g. video-based head-tracking, adaption for virtual table, stereo viewing)
Applications (I)
Applications (II)

- Architectural visualization
- Simulation result visualization
- Planning support
- Seamless integration in WWW courses
- Medical applications
- 3D graphical user interfaces
Conclusion

• Animation element concept is a novel Clipart-like authoring and presentation paradigm
• Cost-efficiency and Reusability
• Modular toolkit
• CASUS: Implementation with VRML output
Question & Answers