



OpenManip:

An Extensible Cross-Scene-Graph Framework for Direct Object Manipulation

Michael Braitmaier
Manfred Weiler
Thomas Ertl

Visualization and Interactive Systems Group
University of Stuttgart

2003-04-02

Michael Braitmaier

1



Overview

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

> Layer Concept

> Extensibility

> Integration

> Results

> Conclusion
2003-04-02

- ⇒ What is a manipulator?
- ⇒ Motivation
- ⇒ Core Components
- ⇒ Event Handling
- ⇒ Layer Concept
- ⇒ Extensibility
- ⇒ Integration Into Applications
- ⇒ Results
- ⇒ Conclusion

Michael Braitmaier

2



What is a Manipulator?

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

> Layer Concept

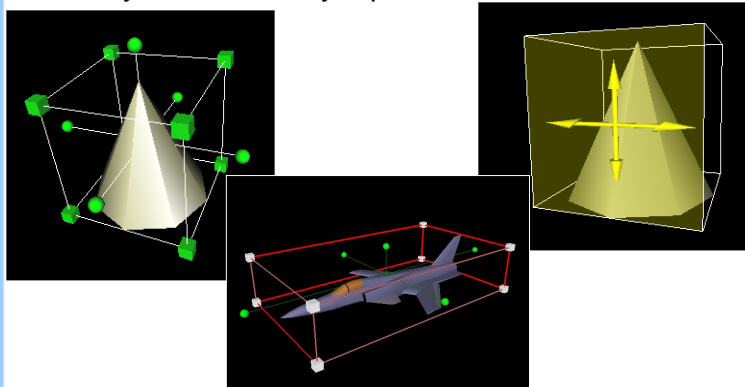
> Extensibility

> Integration

> Results

> Conclusion
2003-04-02

- ⇒ Tool for direct object manipulation
- ⇒ Early introduced by OpenInventor



Michael Braitmaier

3



Motivation

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

> Layer Concept

> Extensibility

> Integration

> Results

> Conclusion
2003-04-02

- ⇒ OpenInventor
 - ⇒ No free manipulator library
 - ⇒ Less suitable for high-performance VR applications
- ⇒ Which scene graph API will prevail?
 - ⇒ OpenInventor
 - ⇒ Performer
 - ⇒ Cosmo3D
 - ⇒ Fahrenheit
- ⇒ Support a wide variety of scene graph APIs
- ⇒ Extendable manipulators

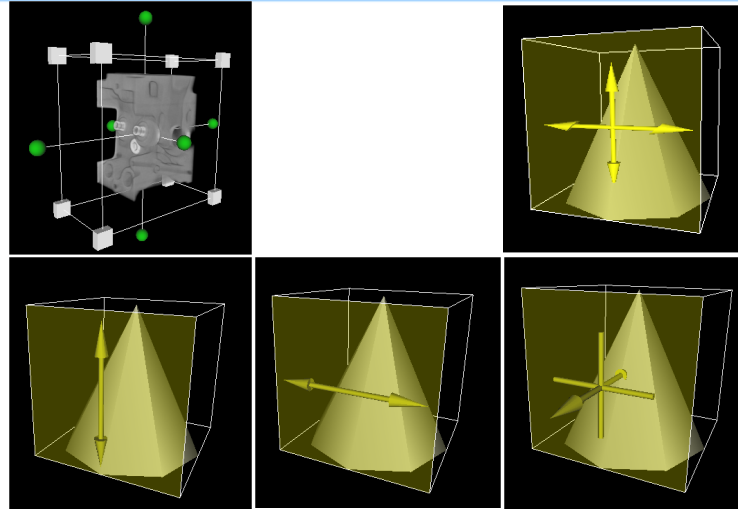
Michael Braitmaier

4



How a manipulator works (1)

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion 2003-04-02

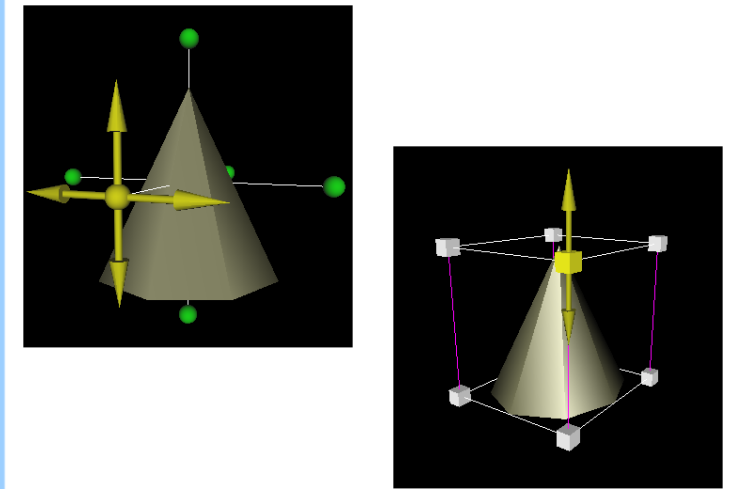


Michael Braitmaier



How a manipulator works (2)

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion 2003-04-02



Michael Braitmaier

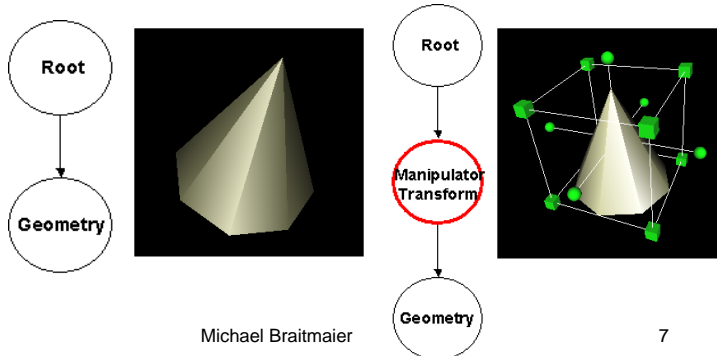


Manipulators

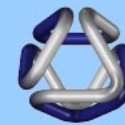
- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion 2003-04-02

Manipulators

- ⇒ Inserts/removes transform into scene graph
- ⇒ Provide data management
- ⇒ Consist of one to (n) Dragger



Michael Braitmaier

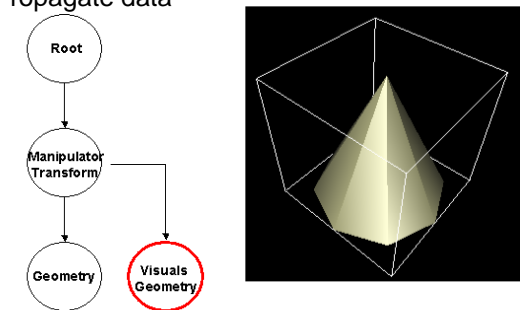


Draggers

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion 2003-04-02

Draggers

- ⇒ Realizes a single transform type
- ⇒ Manage Handles and Visuals
- ⇒ State switching
- ⇒ Propagate data



Michael Braitmaier



Handles

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

> Layer Concept

> Extensibility

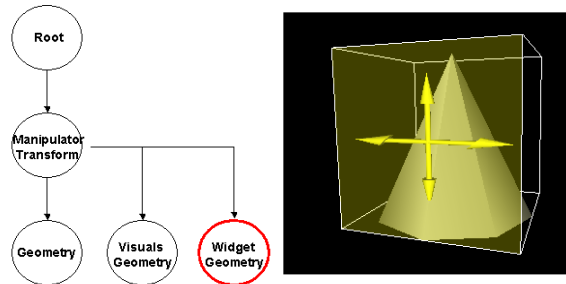
> Integration

> Results

> Conclusion
2003-04-02

⇒ Handles

- ⇒ Provide widgets for interaction
- ⇒ Maintain state changes
- ⇒ Perform transformation calculation



Michael Braitmaier

9



Projectors

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

> Layer Concept

> Extensibility

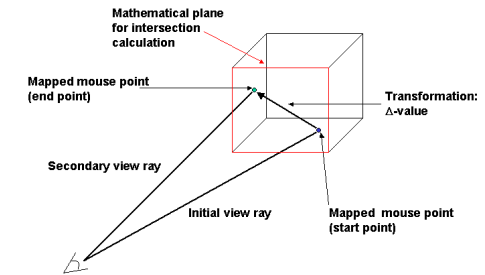
> Integration

> Results

> Conclusion
2003-04-02

⇒ Projectors

- ⇒ Map mouse movement
- ⇒ Pure analytical determination



Michael Braitmaier

10



Component Interaction

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

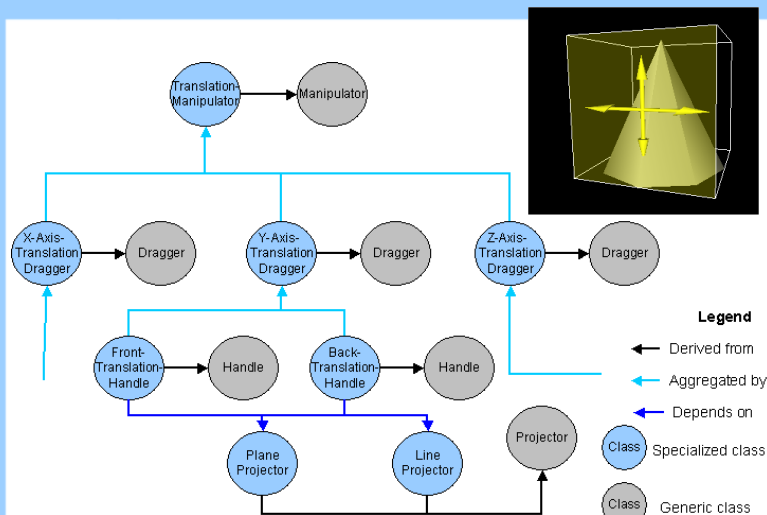
> Layer Concept

> Extensibility

> Integration

> Results

> Conclusion
2003-04-02



Michael Braitmaier

11



Event Handling (1)

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

> Layer Concept

> Extensibility

> Integration

> Results

> Conclusion
2003-04-02

⇒ EventDistributor

- ⇒ Receives events
- ⇒ Checks for affected Handles
- ⇒ Forwards events

⇒ EventListener

- ⇒ Abstract class to provide event receiving ability

⇒ Callback list

- ⇒ Callback objects contain
 - Node associated with Handle
 - Reference to Handle (EventListener)

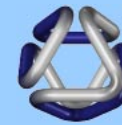
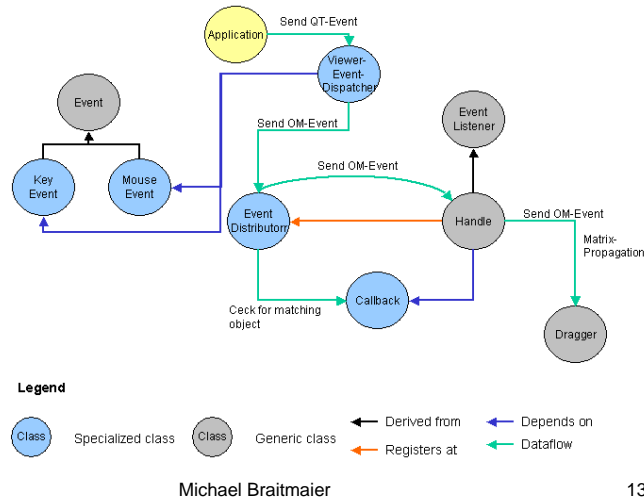
Michael Braitmaier

12



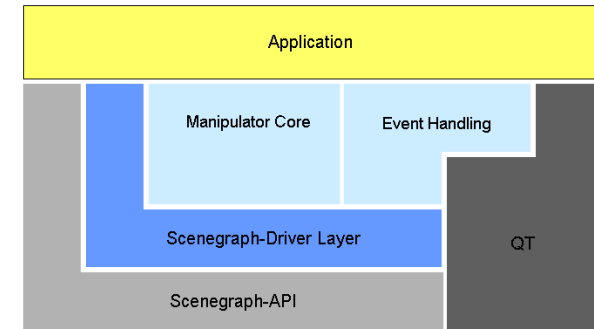
Event Handling (2)

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02



The OpenManip Layer Concept

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02



- ⇒ Layer for manipulator core and event handling
- ⇒ Layer for scene graph driver

Michael Braitmaier 14



Scene Graph Driver Layer (1)

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ⇒ Scene graph abstraction layer providing
 - ⇒ Classes with generic functionality
 - ⇒ Stable interface
- ⇒ Wraps following scene graph parts
 - ⇒ Basic geometry
 - ⇒ Lights
 - ⇒ Transform and group nodes
 - ⇒ Camera
- ⇒ Wraps scene graph API peculiarities

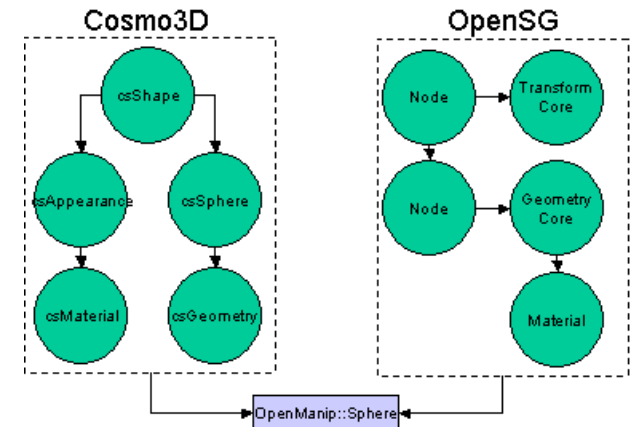
Michael Braitmaier 15



Scenegraph Driver Layer (2)

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ⇒ Example: Sphere



Michael Braitmaier 16



OpenSG Implementation

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ➔ No event handling in OpenSG
- ➔ No specialized primitives in OpenSG
 - ➔ Object modifications expensive
 - ➔ Additional transform nodes
- ➔ Picking through BaseActions
- ➔ Node/NodeCore concept eased implementation
- ➔ Required functionality well supported
- ➔ Excessive coding

Michael Braitmaier

17



Cosmo3D Implementation

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ➔ Provides no consistent class hierarchy
 - ➔ E.g. Nodes vs. Geometries
- ➔ Picking provided by csCamera object
- ➔ Provides basic primitives
 - ➔ Methods for easy manipulation
 - ➔ Transform not necessarily required

Michael Braitmaier

18



Extensibility

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ➔ Two types of extension
 - ➔ Scene graph driver extension: new scenegraph driver for additional scene graph APIs
 - Rewrite about 30 classes
 - Respect scene graph peculiarities
 - ➔ Toplayer extension: new manipulators, draggers

Michael Braitmaier

19



What OpenManip does for you?

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ➔ Management of data exchange
- ➔ Automatic deletion of Visuals and Handles
- ➔ Propagation of changes
 - ➔ E.g. changes of the transformation matrix
- ➔ Event handling and event distribution

Michael Braitmaier

20



What remains to be done?

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ➔ Derive new manipulator from generic class
- ➔ Main functionality provided by Handles
- ➔ Write code for
 - ➔ Widget presentation
 - ➔ State changes
 - ➔ Incorporation of Projectors
- ➔ Register Handles EventDistributor
- ➔ Effort correlates with complexity of new manipulator

Michael Braitmaier

21



Integration Into Applications

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ➔ Only 3 basic objects required
 - ➔ Scenegrph
 - ➔ EventDistributor
 - ➔ ViewerEventDispatcher
- ➔ Redirect event handling to ViewerEventDispatcher
- ➔ Maintain a list of active manipulators

Michael Braitmaier

22



Results (1)

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02

- ➔ OpenManip supports two scene graphs
 - ➔ OpenSG
 - ➔ Cosmo3D
- ➔ OpenManip is integrated into two applications
 - ➔ One special demo application viewer
 - ➔ Volume viewer application

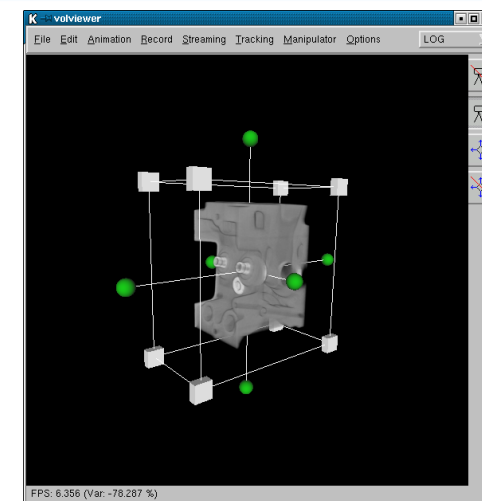
Michael Braitmaier

23



Results Screenshots(1)

- > Overview
- > What is a manipulator?
- > Motivation
- > Core Components
- > Event Handling
- > Layer Concept
- > Extensibility
- > Integration
- > Results
- > Conclusion
2003-04-02



Michael Braitmaier

24



Results (2)

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

> Layer Concept

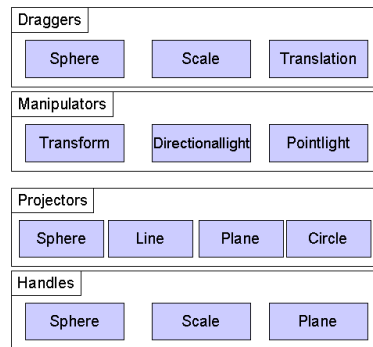
> Extensibility

> Integration

> Results

> Conclusion
2003-04-02

⇒ OpenManip library currently provides



Michael Braitmaier

25



Results Screenshots(2)

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

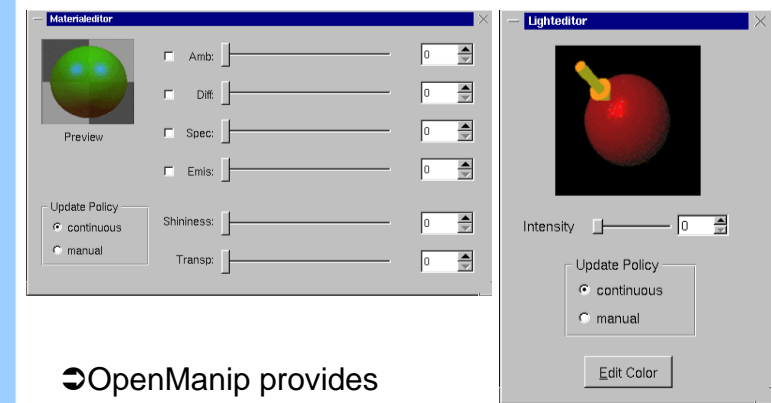
> Layer Concept

> Extensibility

> Integration

> Results

> Conclusion
2003-04-02



⇒ OpenManip provides

⇒ Qt-based material editor

⇒ Qt-based headlight editor

Michael Braitmaier

26



Conclusion

> Overview

> What is a manipulator?

> Motivation

> Core Components

> Event Handling

> Layer Concept

> Extensibility

> Integration

> Results

> Conclusion
2003-04-02

- ⇒ OpenManip framework for direct object manipulation
- ⇒ Inspired by OpenInventor manipulators
- ⇒ Extensibility ensured by object oriented layer concept
 - ⇒ Manipulators
 - ⇒ Scene graph drivers
- ⇒ Cross-scene-graph framework
 - ⇒ Abstraction from scene graph APIs
 - ⇒ Support for Cosmo3D, OpenSG
- ⇒ Low effort for integrating OpenManip

Michael Braitmaier

27