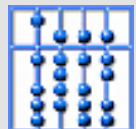# Design Patterns for Augmented Reality Systems

MIXER 2004

Asa MacWilliams, Thomas Reicher,

Gudrun Klinker, Bernd Bruegge

Lehrstuhl für Angewandte Softwaretechnik
Institut für Informatik, Technische Universität München

macwilli@in.tum.de

Dec 19 2003

Summary

- *Patterns* describe reusable problem-solving knowledge

- We propose to develop a collection of *software patterns for Augmented Reality systems*

- These patterns are classified by subsystems

- Patterns are described using a standard *scheme*

- Patterns are interrelated, forming a *system* of patterns

- To be successful, this must be a joint community effort
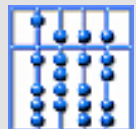
Motivation & Methods

- Most current AR systems are ad hoc solutions
- However, reusable problem-solving knowledge exists in the construction of software for AR systems
- Patterns have successfully captured such knowledge
  - Design patterns (Gamma et al.): Observer, Adapter, Bridge…
  - Architectural patterns (Buschmann et al.): Model/View/Controller…
- A system of software patterns for AR can:
  - aid architects of new AR systems
  - enable comparisons between existing AR systems
  - facilitate sharing design experience between research groups
- How to do this?
  - Extract common approaches from many different AR systems
  - Abstract these approaches to form patterns
  - Discuss the patterns within the AR community
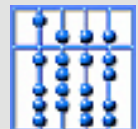
Definition

A software architectural pattern

for Augmented Reality systems *describes*

- *a specific design problem for a particular AR subsystem*

- *which appears in a particular design context,*

- *and presents a generic solution scheme.*
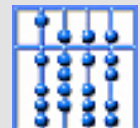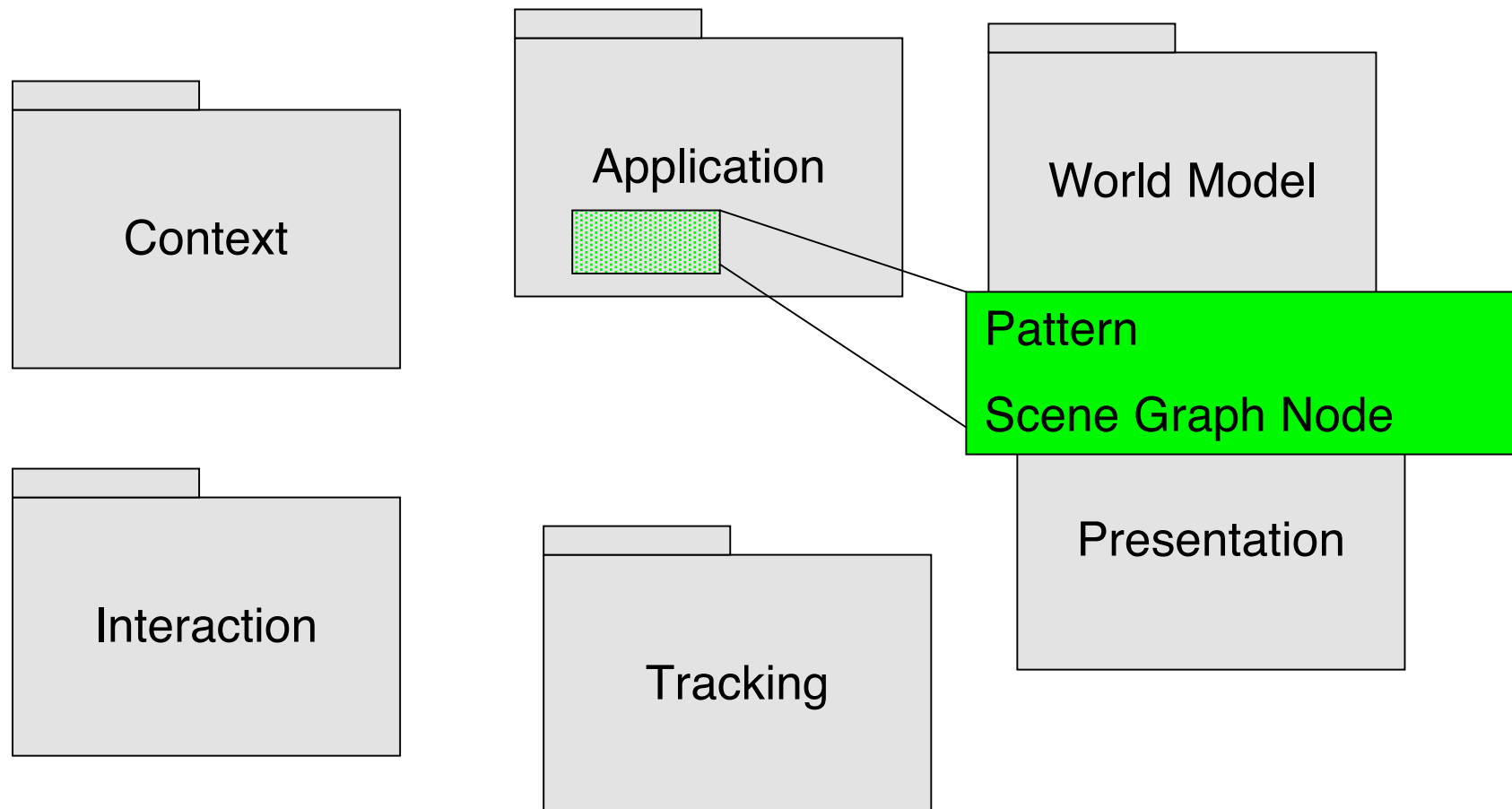
The solution scheme specifies

- *the involved components,*

- *their responsibilities,*

- *relationships*

- *and the way they cooperate.*

[**Specialization** of Buschmann et al.]

Augmented Reality Subsystems

Simplified Organization of many AR Systems

Example Pattern Description (abbrev.)

Name: Scene Graph Node (Application)

Goal: Embed application in scene graph.

Motivation: In AR, user interaction is connected with the spatial environment. With this approach, the application is seamlessly embedded in the environment.

Description: A scene graph models the world around a user as a tree of nodes. Each node can be any type object, usually graphical ones. But there are also nongraphical objects that include control code.

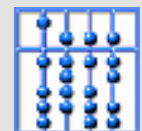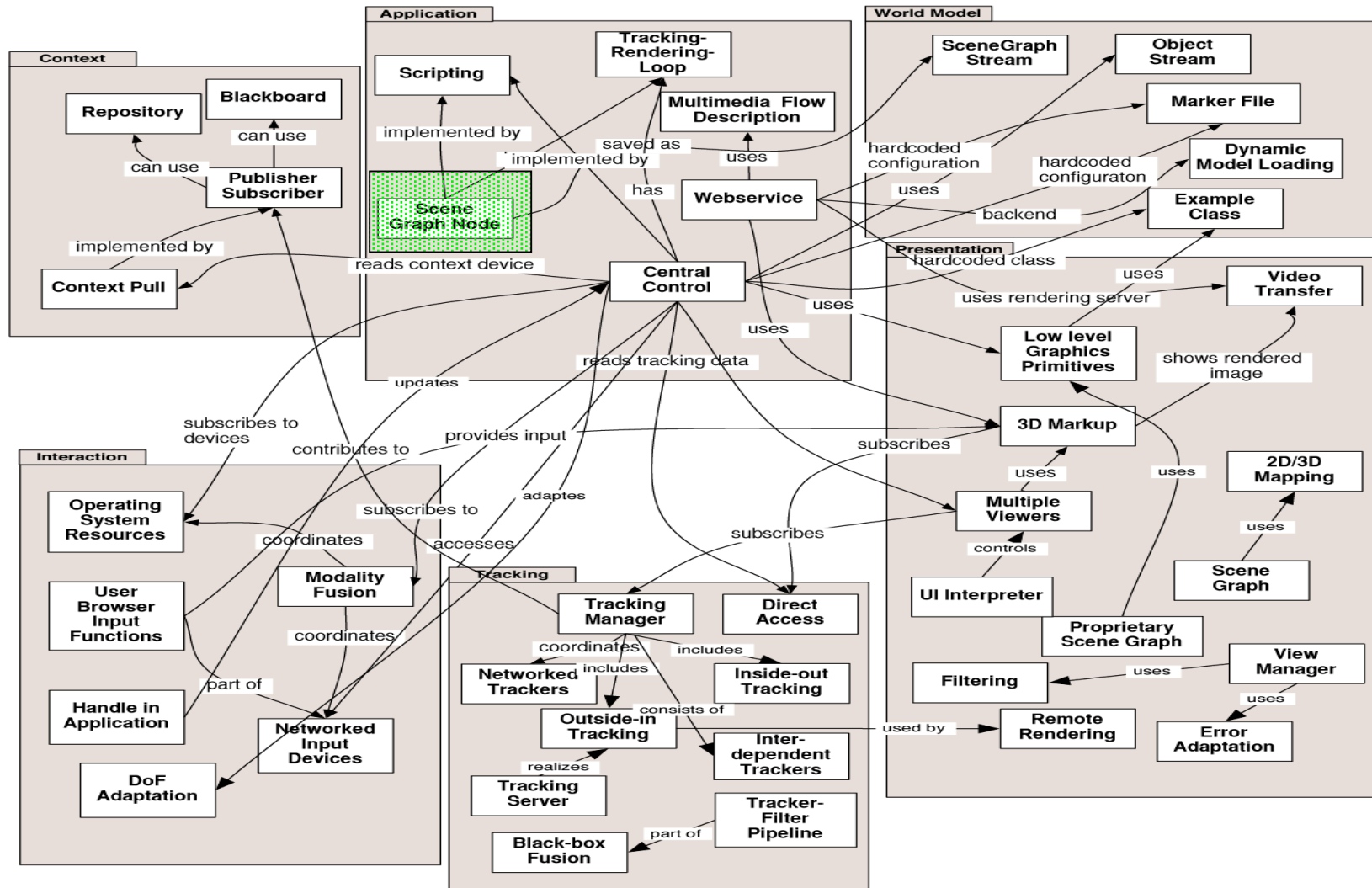**Usability:** Requires *scene graph*-based renderer.

Consequences: The Scene Graph Node pattern handles the control flow to the underlying scene graph platform. Using scene graph replication, this offers an easy way for the implementation of shared applications for locally nearby users. The 3D interface can be shared among several users but displayed for each from a different view.

Collaboration: Requires *scene graph* presentation pattern; may be implemented with using *scripting*.

Known use: Studierstube, Tinmith, MARS
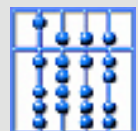
A system of patterns

The current collection of patterns

Application Subsystem: Central Control, Multimedia Flow Description, Scene Graph Node, Scripting, Tracking-Rendering-Loop, Web Service
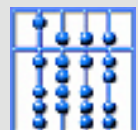
Context Subsystem: Blackboard, Context Pull, Publisher/Subscriber, Repository

Interaction Subsystem: DOF Adaption, Handle In Application, Modality Fusion, Networked Input Devices, Operating System Resources, Use Browser Input

Presentation Subsystem: Adaption To Error Level, Low-Level Graphics Primitives, Multiple Viewers, Presentation Filter, Proprietary Scene Graph, Remote Rendering, Scene Graph, 3D Markup, 2D/3D Mapping, User Interface Interpreter, Video Transfer, View Manager

Tracking Subsystem: Black-Box Fusion, Direct Access, Inside-Out Tracking, Interdependent Trackers, Networked Trackers, Outside-In Tracking, Tracker-Filter Pipeline, Tracking Manager, Tracking Server

World Model Subsystem: Dynamic Model Loading, Example Class, Marker File, Object Stream, Scene Graph Stream

The current collection of patterns

Application Subsystem: Central Control, Multimedia Flow Description, Scene Graph Node, Scripting, Tracking-Rendering-Loop, Web Service

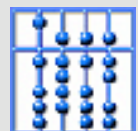**Context Subsystem**: Blackboard, Context Pull, Publisher/Subscriber, Repository
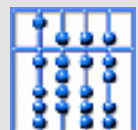
Interaction Subsystem: DOF Adaption, Handle In Application, Modality Fusion, Networked Input Devices, Operating System Resources, Use Browser Input

Presentation Subsystem: Adaption To Error Level, Low-Level Graphics Primitives, Multiple Viewers, Presentation Filter, Proprietary Scene Graph, Remote Rendering, Scene Graph, 3D Markup, 2D/3D Mapping, User Interface Interpreter, Video Transfer, View Manager
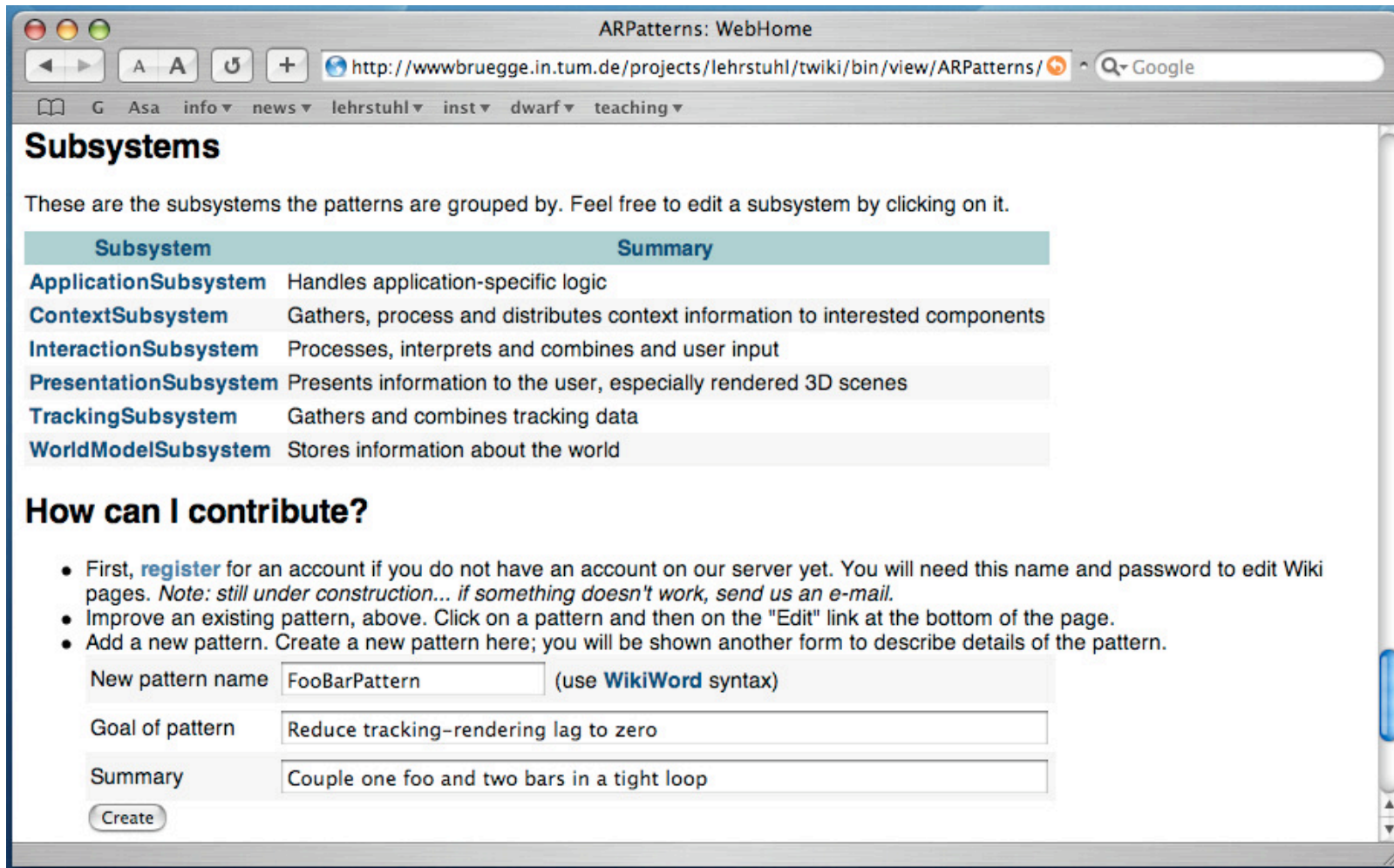
Tracking Subsystem: Black-Box Fusion, Direct Access, Inside-Out Tracking, Interdependent Trackers, Networked Trackers, Outside-In Tracking, Tracker-Filter Pipeline, Tracking Manager, Tracking Server

World Model Subsystem: Dynamic Model Loading, Example Class, Marker File, Object Stream, Scene Graph Stream

… Patterns in *ARVIKA*

The current collection of patterns

Application Subsystem: Central Control, Multimedia Flow Description, Scene Graph Node, Scripting, Tracking-Rendering-Loop, Web Service

Context Subsystem: Blackboard, Context Pull, Publisher/Subscriber, Repository

Interaction Subsystem: DOF Adaption, Handle In Application, Modality Fusion, Networked Input Devices, Operating System Resources, Use Browser Input

Presentation Subsystem: Adaption To Error Level, Low-Level Graphics Primitives, Multiple Viewers, Presentation Filter, Proprietary Scene Graph, Remote Rendering, Scene Graph, 3D Markup, 2D/3D Mapping, User Interface Interpreter, Video Transfer, View Manager

Tracking Subsystem: Black-Box Fusion, Direct Access, Inside-Out Tracking, Interdependent Trackers, Networked Trackers, Outside-In Tracking, Tracker-Filter Pipeline, Tracking Manager, Tracking Server

World Model Subsystem: Dynamic Model Loading, Example Class, Marker File, Object Stream, Scene Graph Stream

**… Patterns in *AR Toolkit "Simple Test"***

How can I contribute?

www.augmentedreality.de -> „Software Patterns for AR"
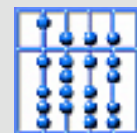
Discussion

- **Where are we now?**
  - We have a rudimentary collection of patterns on our web site
  - We have a classification and description scheme for the patterns

- **Open questions:**
  - What about patterns involving more than one subsystem?
  - How to describe patterns of different levels of sophistication?

- **What should happen next?**
  - Improve description of patterns with input from the experts, i.e. the systems' architects
  - Condense list, focusing on "most successful" patterns
  - Test patterns, e.g. by building new AR systems based on them

- **What are key factors for success?**
  - "Critical mass" of AR researchers joining in
  - Agreed-upon terminology, crucial for a communication medium
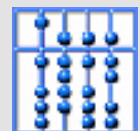
Design Patterns for Augmented Reality Systems

MIXER 2004

Asa MacWilliams, Thomas Reicher,

Gudrun Klinker, Bernd Bruegge

Thank You for Your Attention!

Any Questions?

macwilli@in.tum.de