Results of a Study on Software Architectures for Augmentd Reality Systems

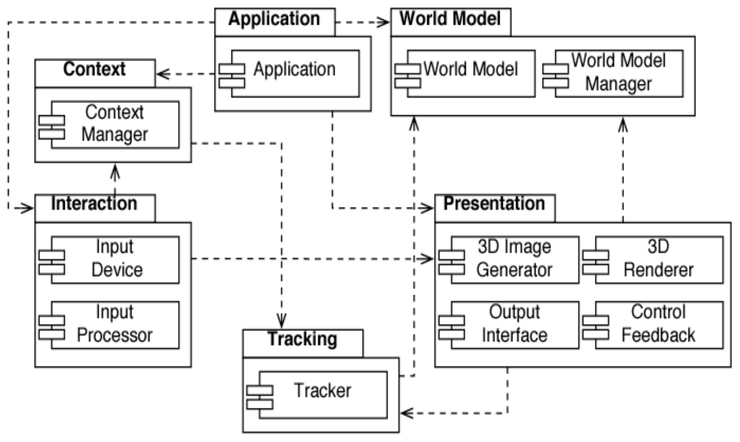
Thomas Reicher, Asa MacWilliams, Bernd Bruegge, Gudrun Klinker
Chair for Applied Software Engineering
Institut für Informatik, Technische Universität München
Boltzmannstr. 3, Garching bei München, Germany
(reicher,macwilli,bruegge,klinker)@in.tum.de

Introduction

Most existing Augmented Reality (AR) systems focus on a particular subsystem, such as position tracking or human-computer interaction. Only a few take a comprehensive approach with AR as part of an enterprise-wide system. An overview of this study is presented here; the full version is available from the authors upon request.

For the ARVIKA consortium, we conducted a study on AR soft-ware architectures, analyzing ARVIKA, AIBAS, AR-PDA, Aura, ARToolkit, ArcheoGuide, BARS, the Boeing wire bundle assembly prototype, DWARF, EMMIE, ImageTclAR, MARS, MR Platform, prototypes by Siemens Corporate Research, STAR, Studierstube, Tinmith, and UbiCom. In order to facilitate comparison of AR software architec-tures described in different notations, we extracted a *refe-rence architecture* and standard terms for software com-ponents typically found in AR systems. We can, to some extent, map existing architectures onto it.

Augmented Reality Reference Architecture



We identified six subsystems common to most AR architectures:

Application, containing application-specific logic and content, and access to legacy systems and databases;

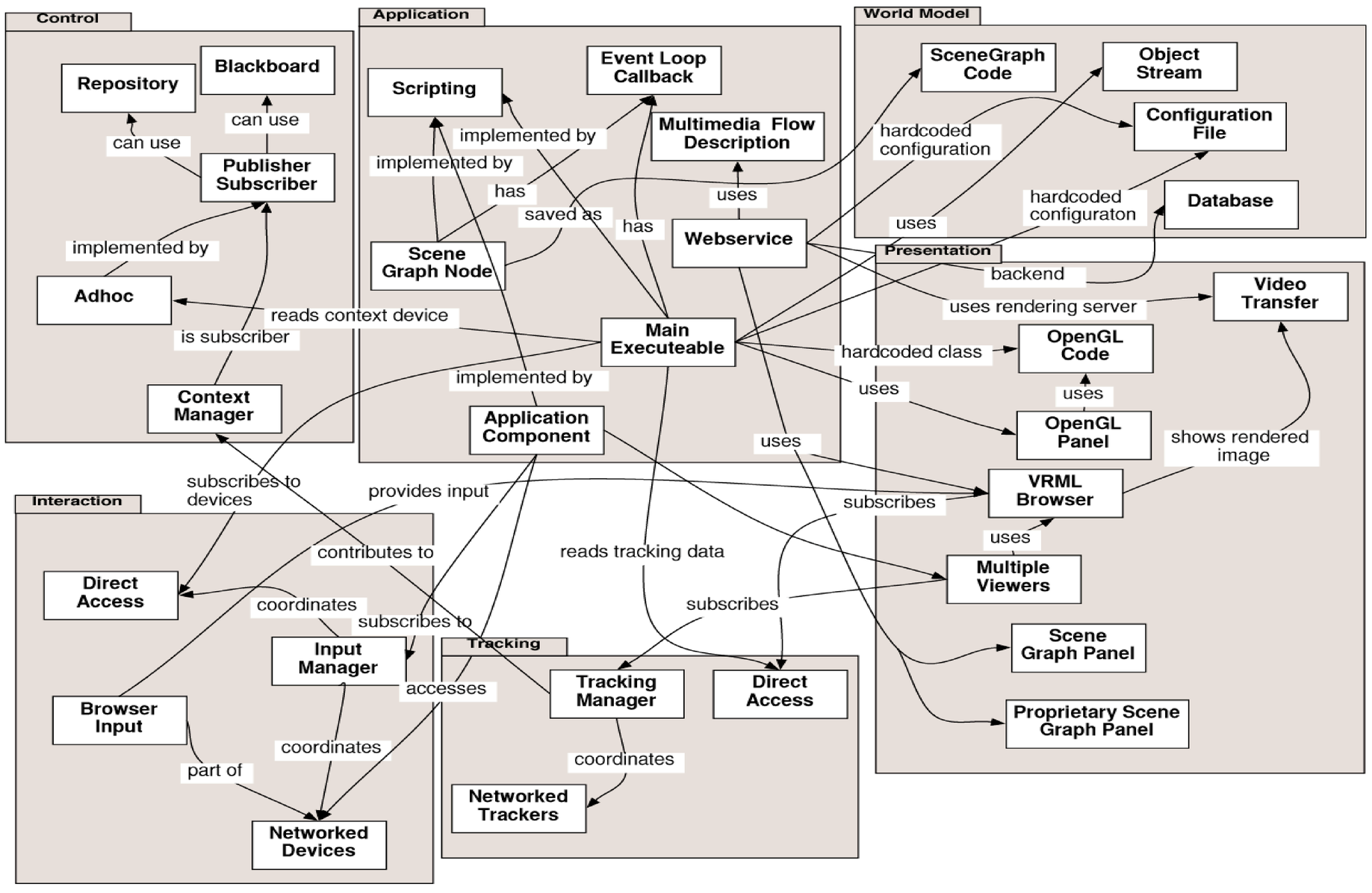
Tracking, responsible for determining the users' and other objects' pose;

Interaction, which gathers and processes user input;

Presentation, which uses 3D and other output modalities;

Context, which collects different types of con-text data and makes it available to other subsystems; and

World Model, which stores and provides information about real and virtual objects around the user.



Software Patterns for Augmented Reality

Based on the reference model, we identified several commonly used *patterns* for implementing them. We see this as a first step towards the development of a pattern language for AR systems in the sense of the design pattens of Gamma et al. We cataloged the approaches, providing goal, motivation, description, usability, consequences and known use for each. This allows developers to consider the impact of design decisions on the system's quality attribtues. The figure above shows the relationships between the approaches we have identified.

The identified pattern set is not yet complete.

We invite you to add your own patterns!

Exemplary Pattern Description

Name: Scene Graph Node (Application)

Goal: Embed application in scene graph.

Motivation: In AR, user interaction is connected with the physical 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: In combination with a scene graph-based renderer.

Consequences: The Scene Graph Node pattern handles the control flow to the underlying scene graph platform. 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.

Known use: Studierstube, Tinmith, MARS

This work was supported by the German Federal Ministry for Education and Research (BMBF) in the ARVIKA project. We are obliged to the ARVIKA members for support of this study. We would like to thank all AR researchers who filled out our questionnaire, providing the all-important raw data for our evaluation; and the members of the Software Engineering Institute, who provided helpful insights into our architecture evaluation methods.

DWARF
Distributed Wearable Augmented Reality Framework
www.augmentedreality.de