Results of a Study on Software Architectures for Augmented Reality Systems

Thomas Reicher, Asa MacWilliams, Bernd Brügge, and Gudrun Klinker
Institut für Informatik, Technische Universität München
Boltzmannstraße 3, Garching bei München, Germany
(reicher,macwilli,bruegge,klinker)@in.tum.de

Abstract

Research prototypes in AR usually do not emphasize software architecture. Nevertheless, their architecture is not arbitrary, but results from specific needs. Architectural approaches embodying research contributions are of particular value for reuse at component and architectural levels.

We have conducted a study of existing AR software architectures for the ARVIKA project [10]. This has resulted in a catalog of important desired quality attributes for AR systems, a reference architecture for comparison of AR architectures, and a catalog of architectural approaches used in current AR systems. We believe this lays the foundation for further research in AR software architectures.

1. 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. The link to enterprise information systems, an important aspect in industry, is rarely considered.

Within the ARVIKA consortium, we conducted a study on AR software architectures, analyzing ARVIKA [10], AIBAS [15], ArcheoGuide [1], AR-PDA [2], AR-Toolkit [3], Aura [4], BARS [5], the Boeing wire bundle assembly prototype [7], DWARF [8], EMMIE [9], ImageTclAR [12], MARS [13], MR Platform [14], prototypes by Siemens Corporate Research [16], STAR [17], Studierstube [18], Tinmith [19], and UbiCom [20].

An overview of this study is presented here; the full version is available from the authors upon request.

2. Desired Quality Attribtues

In questionnaires we distributed to many different AR research groups, we asked which *quality attributes* were important considerations for the architecture, following methods from the Architecture Tradeoff Analysis Method [6].

Among the architecturally relevant *high-priority* quality attributes were tracking and rendering latency, wireless and network-disconnected operation, use of multiple tracking devices, component addition and reuse, and the ability to integrate existing AR components.

Several *low-priority* quality attributes were limiting CPU load, fault tolerance and system uptime, security, reconfigurability at runtime, support for different simultaneous input modalities, adaptability to users' preferences and abilities, support for multiple users, support for multiple hardware platforms, and ease of integrating legacy components.

3. Reference Architecture

In order to facilitate comparison of AR software architectures described in different notations, we extracted a *reference architecture* in which we define standard terms for software components typically found in AR systems (Figure 1). While no single system actually uses this architecture, each can, to some extent, be mapped onto it.

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; *Control*, which gathers and processes user input; *Presentation*, which uses 3D and other output modalities; *Context*, which collects different types of context data and makes it available to other subsystems; and *World Model*, which stores and provides information about real and virtual objects around the user.

We have mapped ARVIKA and DWARF onto this reference architecture; we propose to map others in the future.

4. Architectural Approaches

Based on the reference model, we identified several commonly used *approaches* for implementing them. We see this

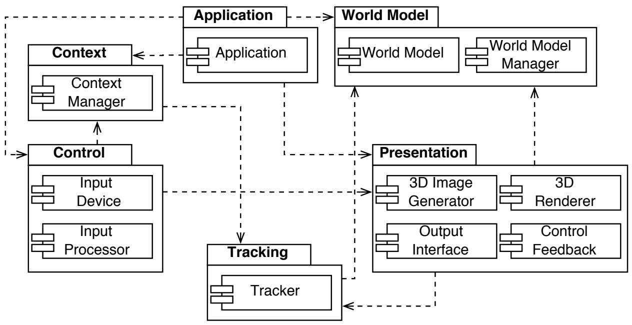

Figure 1. Reference architecture showing subsystems and their dependencies

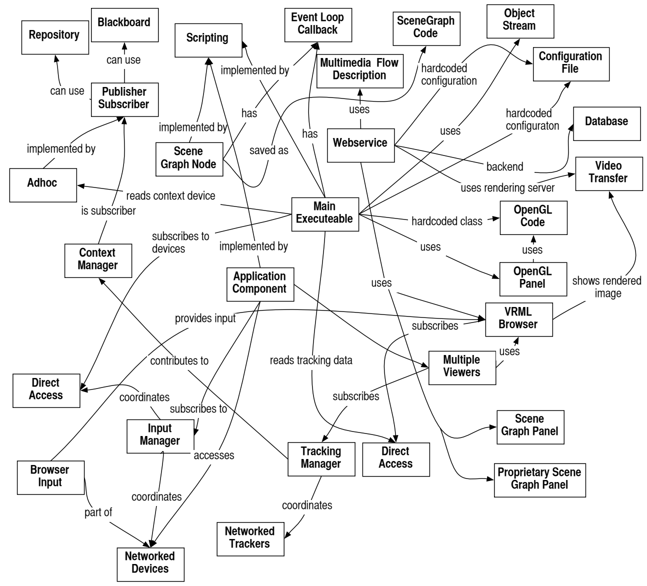

Figure 2. Relationships between approaches for subsystem implementation

as a first step towards the development of a pattern language for AR systems in the sense of [11]. We catalogued 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 attributes. Figure 2 shows the relationships between the approaches we have identified.

5. Conclusion

In this work, we have laid the foundation for further research in AR software architectures. We would like to encourage the AR community to compare their software architectures with our reference architecture, and to extend our catalog of architectural approaches (on the web at [8]), especially for the more difficult areas of AR. We believe that, in the future, this could result in a guidebook towards designing application-specific AR software architectures, providing a substantial aid to AR software architects.

Acknowledgements

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.

References

[1] `archeoguide.intranet.gr/project.htm`.
[2] `www.ar-pda.com/index_e.htm`.
[3] `www.hitl.washington.edu/projects/shared_space/download`.
[4] `www-2.cs.cmu.edu/~aura/`.
[5] `ait.nrl.navy.mil/vrlab/projects/BARS/BARS.html`.
[6] P. Clements, R. Kazman, and M. Klein. *Evaluating Software Architectures: Methods and Case Studies*. Addison Wesley, Boston, MA, 2002.
[7] D. Curtis, D. Mizell, P. Gruenbaum, and A. Janin. Several devils in the detail: Making an ar app work in the airplane factory. In *Proceedings of IWAR '98*, Nov. 1998.
[8] `www.augmentedreality.de`.
[9] `www.cs.columbia.edu/graphics/projects/emmie/emmie.html`.
[10] W. Friedrich, D. Jahn, and L. Schmidt. Arvika - augmented reality for development, production and service. Proceedings of the International Status Conference HCI, 2001.
[11] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1995.
[12] `metlab.cse.msu.edu/imagetclar/`.
[13] `www.cs.columbia.edu/graphics/projects/mars/mars.html`.
[14] `www.mr-system.co.jp/mr-platform/`.
[15] C. Robertson and B. MacIntyre. Adapting to registration error in an intent-based augmentation system. In *ACM User Interface Software and Technology 2002 (UIST 2002)*, Paris, France, Oct. 2002. Presented as a poster.
[16] `www.scr.siemens.com/2a.html`.
[17] `www.realviz.com/STAR/`.
[18] `www.studierstube.org`.
[19] `www.tinmith.net`.
[20] `www.ubicom.tudelft.nl`.