

Software Development Challenges for Ubiquitous Augmented Reality

Asa MacWilliams

Technische Universität München, Fakultät für Informatik
Boltzmannstraße 3, 85748 Garching bei München, Germany
Tel.: +49 89 289 18228, Fax: +49 89 289 18207
macwilli@in.tum.de

Abstract

The research areas of augmented reality and ubiquitous computing both contribute to a more natural interaction of users with computing systems in their environment. As each area matures, they are beginning to overlap. Their confluence is the domain for my dissertation, and I call it *ubiquitous augmented reality*.

Building ubiquitous augmented reality systems presents three software engineering challenges. First, the software must cope with uncertainty; the users' mobility changes the availability of distributed devices. Second, the desired behavior of a system is ill-defined, as appropriate interaction metaphors are still being researched and users' preferences change. Third, the system must maintain near-real-time performance to create a convincing user experience.

In this position paper, I present the domain of ubiquitous augmented reality, show several example applications, and describe the software engineering challenges. These serve as a motivation for ongoing and future research.

1 Introduction

Both augmented reality and ubiquitous computing contribute to a more natural interaction of users with computing systems in their environment, letting them access information in a real-world context. The combination of these paradigms enables the development of new applications that are not possible separately. As each research area matures, they are beginning to overlap. Their confluence is the domain for my research, and I call it *ubiquitous augmented reality*, or UAR.

In my forthcoming dissertation [Mac03], I propose to address several challenges of developing software for ubiquitous augmented reality, with a distributed, adaptive software architecture and an incremental development process. In this position paper, I give a definition of UAR, present example applications to show the relevance of the domain, describe three key challenges in software development, and suggest architectural directions in which these challenges can be addressed.

2 The Domain of Ubiquitous Augmented Reality

What is ubiquitous augmented reality? The area can be defined based on classic definitions of augmented reality and ubiquitous computing.

Augmented reality – or AR – as defined by Azuma in [Azu97] –

- combines real and virtual,
- is interactive in real time,
- and is registered in three dimensions.

Ubiquitous computing – or ubicomp – as described by Weiser in [Wei93] –

- aims to make many computers available throughout the physical environment,
- aims to make them effectively invisible to the user,
- considers the real world to be wonderful, and aims only to augment it.

As a combination of augmented reality and ubiquitous computing, for the purpose of my research, I define the characteristics of *ubiquitous augmented reality* as follows.

Ubiquitous augmented reality – or UAR –

- augments the real world with virtual information,
- is interactive in real time,
- is spatially registered,
- is available throughout a large physical environment,
- and allows both immersive interaction and unobtrusive assistance.

Thus, UAR combines the distributed, inconspicuous devices of ubiquitous computing and the rich, interactive interfaces of augmented reality. To show that ubiquitous augmented reality is actually relevant, the following section suggests possible applications.

3 Relevance: Applications

While there are many applications that can be implemented with augmented reality alone, or with ubiquitous computing alone, there are several that can benefit from both. These are the applications for ubiquitous augmented reality.

The following examples provide a representative overview of such applications. Each is presented as a brief scenario, followed by challenges in implementing it.

The images included in this section are taken from prototype systems that we have built in our research group; these systems were extremely valuable in discovering the challenges of software development for UAR.

Navigation

A user wishes to reach a specific location on foot. Navigational information is presented on several different devices, depending on the user's preference. This can be head-mounted display (Figure 1), a palmtop computer, or a flat panel screen on the wall in a hallway. The display can include 2D and 3D maps, arrows, path indicators, images of waypoints, and the position of other users [NIH01].

This scenario is challenging to implement if the navigation task is in different environments, e.g. both indoors and outdoors. For different environments, different sensors are needed to determine the user's current position and orientation. Similarly, different types of user interfaces are appropriate in different environments.



Figure 1: Mobile augmented reality: indoor and outdoor pedestrian navigation. Left, “wearable” prototype system; center, outdoor view through head-mounted display when user is looking straight and when looking down; right, view during indoor navigation. (Images from the TRAMP and Pathfinder [BBK⁺01] projects.)

Museum Guide

A visitor freely roams the rooms of a museum, looking at a number of exhibits. To use the UAR museum guide, the visitor wears small, wireless computing equipment with suitable multi-media display facilities, such as headphones, a wrist-worn PDA or a head-mounted display. Depending on the visitor's location, previously seen exhibits, questions asked and preferred interaction modes (audio, video, 2D or 3D graphics, text), the system presents information on the current exhibit. The visitor can request further details interactively, e.g. via a microphone, a dial, or by staring at a particular part of the exhibit for a while.

This scenario is challenging from an interaction point of view, since the user interface must meet certain artistic requirements. An advantage is that it can be installed within the controlled environment of a museum exhibit. On the other hand, the scenario becomes quite complex if users are supposed to bring along their own computing devices, e.g. palmtops, and use them to interact with the exhibit as well.

Multi-Player Games

A commercially promising scenario is a multi-player game with several users operating in a shared environment. The game can range from a first-person shooter (two teams chase each other with virtual weapons) to a collaborative simulated world (Figure 2 on the next page, [MSW⁺03]) or

a distributed campus-wide or city-wide role-playing adventure. The game system has to provide tracking of all users and coordinate their interaction.

The challenges in this application depend on the type of game: a first-person shooter requires fast, low-latency interaction in a distributed wireless network, whereas a distributed adventure game must support users with different sorts of interaction devices.



Figure 2: Augmented reality multi-player shepherding game. Left, virtual sheep floating on hand is shown on head-mounted display and laptop. Center, view through head-mounted display. Right, virtual sheep is scooped off projector table onto palmtop. (Images from the SHEEP [MSW⁺03] project.)

Maintenance

Maintenance of complex technical devices such as cars or aircraft is a promising field for ubiquitous augmented reality [Fri04]. A mobile system assists the mechanic in diagnosing malfunctions and presenting appropriate step-by-step repair instructions (Figure 3).

A challenge of this scenario is that the system must provide different forms of interaction to the user, in order to keep the user's hands free during the maintenance task itself, but also allow scrolling through a manual or data entry. Another challenge is the integration of the mobile system with diagnostic equipment and sensors within the machine itself.



Figure 3: Augmented reality maintenance application. Left, mechanic's view through head-mounted display, showing step-by-step instructions. Right, mechanic with mock-up wearable computer. (Images from the TRAMP project.)

Construction

The construction of complex machines was the original motivation for augmented reality: correctly placing wire bundles in an aircraft. This remains a promising field in industry [Fri04],

especially in the construction of prototypes, where programming a robot would be prohibitively expensive, but unaided manual labor is too slow (Figure 4, [ESK⁺03]). Ubiquitous AR has the potential to extend the range of construction activities, e.g. support an entire building site rather than the production of a single car.

In its challenges, construction is quite similar to maintenance; however, supporting a building site requires the complex combination of many different location tracking sensors.

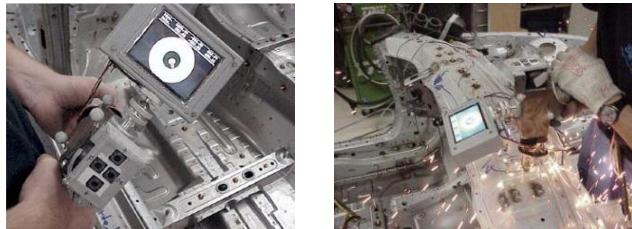


Figure 4: Augmented reality in prototype vehicle construction. Small display on welding gun guides user to the correct 3D welding spot in the construction of a prototype vehicle. (Images from the PAARTI [ESK⁺03] project.)

Intelligent Campus

Students, faculty members and visitors use palmtop or laptop computers and public information displays to locate friends or colleagues, to find lecture halls, or look up schedules. The idea of an intelligent campus is a classic ubiquitous computing application [GBBT03, GSSS01], but can benefit from AR, as well [NIH01], letting users “look through walls” to see where colleagues are. Users collaborate at times, and take advantage of specially instrumented rooms for particularly immersive work.

This scenario is quite challenging, as it requires users to be able to integrate their own devices into the system; it must use location sensors that work both indoors and outdoors; different user interfaces must be provided for laptops and palmtops; and the system’s functionality will evolve significantly over time, as users think of new applications.

Hospital

In a hospital, a ubiquitous augmented reality system could help doctors and nursing staff locate colleagues who are currently on other wards; keep track of the location of patients who are being sent from one treatment room to another; and visualize three-dimensional information, e.g. from CT scans. It could help patients keep track of their treatment schedule, and find their way around the hospital. Thus, the hospital support scenario is similar to the intelligent campus scenario, in that a major feature is displaying the location of people and helping them navigate. Additionally, however, it can take advantage of three-dimensional visualization, to aid doctors in diagnosis, and help explain medical conditions to patients and relatives.

In its challenges, this scenario is similar to the active campus; however, it naturally poses greater security and safety concerns.

Collaborative Design

Virtual reality has been successfully employed in designing many products, such as automobiles, and augmented reality can be used in design as well. It is particularly suitable when exploring early stages of a design, or performing three-dimensional sketches. Ubiquitous augmented reality can extend the designing process even further, by supporting collaborative groups of designers, architects and customers with different kinds of user interfaces in a multi-room or large studio environment (Figure 5). In a studio specially instrumented for 3D architectural modeling, a user with a mobile laptop can walk in and join the modeling task.

Particular challenges in this area involve the design of the user interface, as it must satisfy high aesthetic demands. Also, different input and output devices must be available to explore real and virtual models. If the area in which the design takes place is large (e.g. a building site), then the deployment area becomes a challenge, as well.



Figure 5: Left, automobile design: designer views model of vehicle through head-mounted display. Right, architectural design: virtual building is shown as it appears to an observer through a head-mounted display. (Images from the Fata Morgana [KDB⁺02] and ARCHIE projects).

Team Action

Team action is a scenario where a heterogeneous team can share the capabilities of user-worn devices by offering their services to other users. As an example, a fire brigade has to fight a fire in an office building. Some firemen are outside the building observing the situation, others are inside and can provide more information from the front line. The team action system's task is to automatically collect data such as temperature and the presence of hazardous fumes with devices worn by the firefighters and provide this data to every participating person needing it. This is particularly useful to warn other team members of hazardous areas [JCH⁺04].

This scenario is challenging from an interaction point of view: users are distracted and under pressure; the system should be easy to use. Also, it poses a reliability challenge: Even if part of the system fails (e.g. due to part of a building collapsing), the rest of the system should continue to work as best as possible. Furthermore, the system must support some level of self-organization, so that it can quickly be deployed in a new environment.

Exploration

Archeology, or any other field where the objective is to explore an unknown terrain and annotate it with information, is another application for ubiquitous augmented reality. Archeologists exploring

a site can mark interesting areas with virtual notes, thereby creating maps for themselves and for colleagues. Similarly, an object which is removed from the site can be recorded in three dimensions first, so that subsequent investigators can see it in its original location. This scenario is similar to collaborative design in a wide area; indeed, archeologists can reconstruct historical buildings on the original sites, allowing tourists to see them there virtually [Ioa02].

The challenges in this scenario involve instrumenting a wide area with different types of location sensors and combining different interaction devices, allowing hands-free operation.

4 Software Development Challenges

In developing ubiquitous augmented reality systems, several software engineering challenges must be considered (Figure 6). These affect the *software architecture*, *run-time infrastructure* and *development process*.

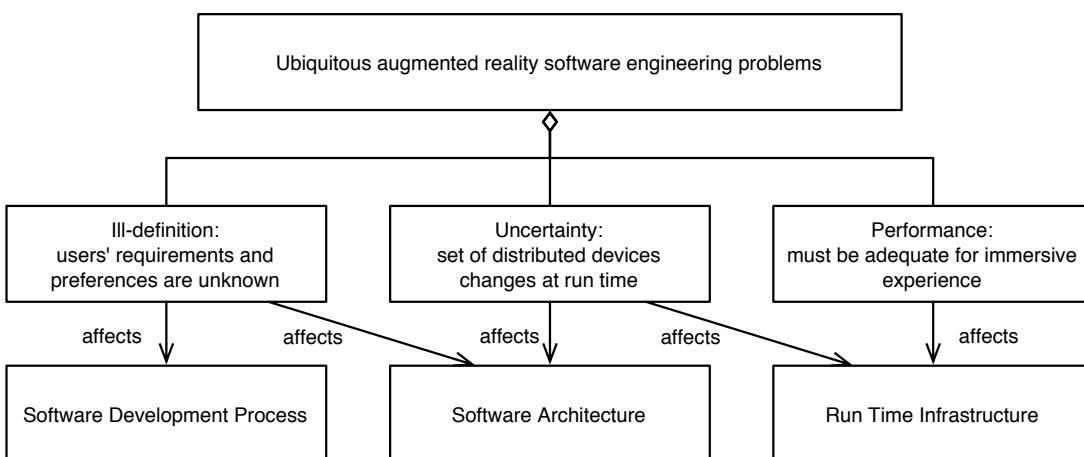


Figure 6: Problems in software development for UAR and their effects on software development.

First, the software must cope with uncertainty; the users' mobility changes the availability of distributed devices, and the users' context changes which combinations of the devices are appropriate. Second, the desired behavior of a system is ill-defined, since users' preferences change, users combine existing functionality in new ways, and appropriate interaction metaphors are still being researched. Third, the system must maintain near-real-time performance to create a convincing user experience.

Note that there are many other challenges of UAR which I do not address, such as developing appropriate abstractions for multimodal user input or modeling different types of distributed tracking sensors. These problems are fairly independent of the software architecture of a UAR system as a whole; and thus are out of the scope of my research.

4.1 Uncertainty

The first challenge is the *uncertainty* that arises when distributed components with changing availability must cooperate in a changing context without knowing each other fully.

Interdependent distributed components A ubiquitous augmented reality system is inherently *distributed*, combining stationary resources such as databases, cameras for position tracking and projection screens with mobile resources such as handhelds, wearable computers and head-mounted displays. Thus, a UAR system must combine distributed hardware and software components.

The components are also *interdependent*: for their correct functionality, components depend on other components, which in turn depend on others and so on. For example, a three-dimensional rendering component in the presentation subsystem depends on up-to-date pose data from one or more trackers, which depend on descriptions of fiducial markers from the world model.

This means that a component does not only depend on other components directly, but, transitively, depends on many other components which it may not be aware of.

Changing availability of devices In ubiquitous computing, and hence in UAR, the sets of devices that are available to form the system changes frequently as users move about. Some devices may simply be beyond reach of a mobile computers' wireless network range, or the network quality is insufficient for the desired type of communication.

Even if we assumed unlimited wireless network coverage, the available network quality is not sufficient for all types of inter-component communication all the time. Furthermore, many devices such as trackers only have a limited physical range of operation; a stationary optical tracker can only operate within one room, since it cannot see through walls.

This means that the architecture of a UAR system cannot be designed to rely on a fixed set of hardware devices; it must *adapt* itself to the available devices.

In general, it cannot even rely on a fixed infrastructure. For example, when two mobile users meet outside, no fixed infrastructure is available, and their computers must form an ad hoc network. An extreme case of this is the *team action* scenario, where a mobile team must be able to cooperate in an unknown environment, and must bring all their required infrastructure with them.

Changing context influences components and system structure The set of hardware devices that should be used in an UAR system, and hence the set of software components, depends on the users' context.

For example, in the *maintenance* scenario, an experienced mechanic may simply wish to glance at the screen of a palmtop fixed to the machine he is working on, whereas an inexperienced mechanic may require detailed instructions in a head-mounted display. Thus, depending on context, the same information should be presented on different output devices.

The user's context can does not only influence the choice of components, but also their communication structure and configuration. For example, when a user enters a room with a stationary optical tracking system, the tracker should be reconfigured to track that user as well. Or, in the *intelligent campus*, when a user leaves his office, an application on his desktop system could "follow"

him on his palmtop.

This means that even once the set of available hardware devices is known, the structure of the communicating software components must still *adapt* itself to the user's context, either automatically or upon explicit user commands. This, again, requires support from the software architecture.

Incomplete knowledge of components In a system that is deployed over a wide area, mobile users will bring new devices and software components into contact with other devices and components that are unknown to the components on their computers.

Thus, a user's mobile computer is confronted with previously unknown software components on other computers, which must be integrated at run time. For example, in the *museum* and *campus* scenarios, users should be able to bring along their own palmtops and use them with the rest of the system.

This means that each software component must be able to operate with incomplete knowledge of the other components. The software architecture must be designed so that each component knows just as much as necessary about the others, and can operate without global knowledge.

Towards a solution To address these challenges, UAR systems need a software architecture that can adapt the set of communicating software components, and their communication structure, to both the availability of devices and to the user's context, in a decentralized fashion [MRB03].

4.2 Ill-definition

As a second challenge, it is hard to specify in advance what the exact behavior of a ubiquitous augmented reality system should be. Many users are involved; they are performing real-world tasks, and the computer system is of secondary interest. The technologies involved are new, and often difficult for users to imagine. Thus, the requirements are *ill-defined*, especially before the system is built and in place.

New and changing technology The possibilities of ubiquitous augmented reality systems are just beginning to be explored. Many usability issues remain to be solved, especially in scenarios such as *team action*, where users are involved in real-world tasks and wish to be supported, not distracted. Promising advances are being made in the area of tracking, especially markerless optical tracking. Hardware is changing rapidly, e.g. in the area of head-mounted displays.

This means that at the current state of research, it is important to be able to build experimental systems quickly, and to improve on them as new results become available.

Ideally, a rapid prototyping environment would make an initial setup for a certain scenario simple to implement, in order to try out new hardware and new human-computer interaction techniques.

Many people involved Due to its wide range of deployment and the collaborative applications, many different people are involved in using, but also in building a ubiquitous augmented reality system.

Not only is the number of people large, but they come from many disciplines: user interface design, human-computer interaction, 3D graphics, sensor analysis, image processing, software development, and application domains such as medicine.

This makes communication between all involved parties about the requirements and behavior of an UAR system difficult, especially before it is built and in place.

New applications In a decentralized system, with many users involved (such as the *campus* scenario), it is impossible to describe, *a priori*, all possible applications the users will find. Users will combine their devices and software components in new ways, creating new individual sub-applications.

As users find new applications, developers will add functionality. Thus the system's functionality is never "finished." If a system is in widespread use, extensions will have to be made while at least parts of the system are running.

During this cycle of identifying new applications and implementing them, entire new classes of applications for a ubiquitous augmented reality system may emerge. Thus, design decisions made when the system was first deployed may no longer be appropriate.

Towards a solution To address these problems, the software architecture must be extensible, allowing a system that has been deployed to be modified afterwards. Furthermore, users and developers should be supported by a development process that allows them to incrementally improve a deployed system, ideally while it is running.

4.3 Performance

Augmented reality has a much higher degree of interactivity and immersivity than ubiquitous computing, and this poses strict performance requirements. Ubiquitous computing systems, however, have a higher degree of distribution than augmented reality systems. Thus, in ubiquitous augmented reality, the tight performance requirements must be met despite a higher degree of distribution and scalability. This is the third challenge.

Immersivity Augmented reality poses stringent requirements on the data flow throughout the system to gain the desired immersion of the user in the augmented environment. To render a three-dimensional scene in a head-mounted display, accurate and timely tracking data on the user's head position and orientation are necessary.

Thus, tracking information must be delivered to presentation components in near real time to be useful. A three-dimensional scene in a head-mounted display must be correctly updated within approximately 30 ms of the user's head movement to prevent motion sickness.

Many communication types Besides positional information, many other data types flow through the system. For example, optical trackers use a stream of video images from a camera in order to determine the position of objects in the camera's view and the camera itself. If the camera is on a

different computer than the one that is performing the image processing, the video image must be sent across the network.

Thus, the system must let components communicate using a variety of different multimedia protocols and formats.

Scalability In the *hospital* and *campus* scenarios, potentially hundreds, if not thousands of users would be using a single ubiquitous augmented reality system. Of course, they tend to collaborate in groups and generally do not all interact with each other simultaneously. Still, the system must be designed so that it scales to a large number of simultaneous users, each of whom is using several computing devices.

Towards a solution To support communication in a distributed UAR system, a common software infrastructure or middleware is required. It must support communication using different protocols, and scale to a large number of computers. At the same time, it must be flexible enough to support an adaptive and extensible architecture; a difficult tradeoff against performance.

5 Conclusion

In this position paper, I have defined the domain of ubiquitous augmented reality and highlighted some of the challenges of developing software for it. Any software architecture, development process or run time infrastructure for UAR will have to address these problems to be successful.

In my dissertation, I plan to address these problems with a software architecture of loosely coupled interdependent services. The architecture deals with uncertainty, using decentralized middleware to dynamically adapt the system's structure. Tools for a dynamic development process let users and developers deal with ill-defined requirements by redesigning the system's behavior at run time. The middleware maintains performance by decoupling communication protocols from system reconfiguration.

In the meantime, I hope that the challenges described in this paper may provide a fruitful basis for discussion in the research community.

References

- [Azu97] Ronald T. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, August 1997.
- [BBK⁺01] Martin Bauer, Bernd Bruegge, Gudrun Klinker, Asa MacWilliams, Thomas Reicher, Stefan Riss, Christian Sandor, and Martin Wagner. Design of a component-based augmented reality framework. In *Proceedings of the International Symposium on Augmented Reality (ISAR)*, October 2001.
- [ESK⁺03] Florian Echtler, Fabian Sturm, Kay Kindermann, Gudrun Klinker, Joachim Stilla, Joern Trilk, and Hesam Najafi. The intelligent welding gun: Augmented reality

for experimental vehicle construction. In S.K Ong and A.Y.C Nee, editors, *Virtual and Augmented Reality Applications in Manufacturing, Chapter 17*. Springer Verlag, 2003.

- [Fri04] Wolfgang Friedrich, editor. *ARVIKA – Augmented Reality für Entwicklung, Produktion und Service*. Publicis, Erlangen, 2004.
- [GBBT03] W. G. Griswold, R. Boyer, S. W. Brown, and T. M. Truong. A component architecture for an extensible, highly integrated context-aware computing infrastructure. In *Proceedings of The International Conference on Software Engineering (ICSE 2003)*, Portland, Oregon, May 2003.
- [GSSS01] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Proactive self-tuning system for ubiquitous computing. In *Proceedings of the Large Scale Networks Conference*, Arlington (VA), March 2001.
- [Ioa02] Nikos Ioannidis. Archeoguide. <http://archeoguide.intranet.gr/project.htm>, 2002.
- [JCH⁺04] Xiaodang Jiang, Nicholas Y. Chen, Jason I. Hong, Kevin Wang, Leila Takayama, and James A. Landay. Siren: Context-aware computing for firefighting. In *Proceedings of the Second International Conference on Pervasive Computing*, Vienna, Austria, April 2004.
- [KDB⁺02] Gudrun Klinker, Allen Dutoit, Martin Bauer, Johannes Bayer, Vinko Novak, and Dietmar Matzke. Fata morgana – a presentation system for product design. In *International Symposium on Augmented and Mixed Reality ISMAR 2002*, 2002.
- [Mac03] Asa MacWilliams. Self-extending systems for context-aware mobile computing. In *Doctoral Symposium at the International Conference on Software Engineering*, Portland, Oregon, USA, May 2003.
- [MRB03] Asa MacWilliams, Thomas Reicher, and Bernd Brügge. Decentralized coordination of distributed interdependent services. In *IEEE Distributed Systems Online – Middleware '03 Work in Progress Papers*, Rio de Janeiro, Brazil, June 2003.
- [MSW⁺03] Asa MacWilliams, Christian Sandor, Martin Wagner, Martin Bauer, Gudrun Klinker, and Bernd Brügge. Herding sheep: Live system development for distributed augmented reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, October 2003.
- [NIH01] Joseph Newman, David Ingram, and Andy Hopper. Augmented reality in a wide area sentient environment. In *Proceedings of the International Symposium on Augmented Reality (ISAR)*, October 2001.
- [Wei93] Mark Weiser. Hot topics: Ubiquitous computing. *IEEE Computer*, Oct. 1993. Three topics: wireless communication, mobility, window systems.