

A Model-based Framework for Context-aware Augmented Reality Applications

Enes Yigitbas, Ivan Jovanovikj, Stefan Sauer
and Gregor Engels

Paderborn University, Fürstenallee 11,
33102 Paderborn, Germany

Enes.Yigitbas@upb.de, Ivan.Jovanovikj@upb.de, Stefan.Sauer@upb.de,
Gregor.Engels@upb.de

Abstract

Augmented Reality (AR) is a technique that enables users to interact with their physical environment through the overlay of digital information. With the spread of AR applications in various domains (e.g. product design, manufacturing or maintenance) and the introduction of concepts such as Pervasive Augmented Reality (PAR), the aspect *context-awareness* started to play an important role. By sensing the user's current context and adapting the AR application accordingly, an adequate user experience can be achieved. Due to the complex structure and composition of AR applications, their development is a challenging task. Although, context-awareness for AR systems was addressed to some extent, a systematic method for development of context-aware AR applications is not fully covered yet. Therefore, in this paper, we identify the main challenges for development of context-aware AR applications and sketch our solution idea for a model-based development framework.

How to cite this book chapter:

Yigitbas, E., Jovanovikj, I., Sauer, S. and Engels, G. 2020. A Model-based Framework for Context-aware Augmented Reality Applications. In: Loizides, F., Winckler, M., Chatterjee, U., Abdelnour-Nocera, J. and Parmaxi, A. (eds.) *Human Computer Interaction and Emerging Technologies: Adjunct Proceedings from the INTERACT 2019 Workshops*. Pp. 303–312. Cardiff: Cardiff University Press. DOI: <https://doi.org/10.18573/book3.am>. License: CC-BY 4.0.

Keywords

Augmented reality · context-awareness

1 Introduction

Augmented Reality (AR) is a user interface metaphor, which allows for interweaving digital data with physical spaces. AR relies on the concept of overlaying digital data onto the physical world, typically in form of graphical augmentations in real-time [1].

Augmented reality has been researched for a considerable amount of time, with first implementations as early as Sutherlands head-mounted three dimensional display “The sword of Damocles” [5] from 1966. The expression *Augmented Reality* was first coined by Tom Caudell in 1992 in his work on the “Application of Heads-Up Display Technology to Manual Manufacturing Processes” [3].

In more recent years, AR technology is strongly on the rise, with many different devices available. One main technology are Head-Mounted-Displays (HMDs) like Microsoft’s HoloLens¹ or the Magic Leap One:² Headsets with integrated display and optics. Some of them also have built in hardware to process the programs that run on the HMD, while other headsets need to be connected to a computer and only serve as a special kind of display which also includes control functions. An alternative way in AR-technology is to use a smartphone as the main hardware. The smartphone can be worn in a headgear (Head-mounted Smartphone), which is not very common for AR applications yet, as many of the headgears only support VR, for example because the phone-camera’s lens is simply covered by the gear. More often smartphones are used in their original purpose, as handheld AR devices.

With the spread and increasing usage of Augmented Reality (AR) techniques in different domains, the need for context-awareness in AR was underlined in previous work [4]. Supporting context-awareness, can greatly enhance user experience in AR applications, for example by adjusting to the individual needs of each user. It also makes the usage more intuitive and effective: The more the application can adjust to the user and his situation, the more natural the AR is experienced and the more ergonomic it is to work with.

However, due to the complex structure (tasks, scenes) and composition (interrelations between real and virtual information objects) of AR applications [6], the development of context-aware AR applications is a challenging task. While context-aware AR applications were introduced for specific application domains, e.g. maintenance [8], a systematic method for supporting the efficient development of context-aware AR applications is not fully covered yet. Therefore,

¹ <https://www.microsoft.com/en-CY/hololens>.

² <https://www.magicleap.com/magic-leap-one>.

in this paper, we discuss the main challenges in developing context-aware AR applications and sketch a first solution idea for a model-based development framework for context-aware augmented reality applications.

The rest of the paper is structured as follows: In Section 2, we discuss main challenges in developing context-aware AR applications. In Section 3, we present architectural patterns as basic solution concepts for addressing these challenges. Section 4 provides an overview of our integrated model-based framework supporting the development of context-aware AR applications. Finally, Section 5 concludes our work with an outlook on future work.

2 Challenges

The challenges in developing context-aware AR applications can be divided up in to three main categories: multi-platform capability, adaptation capability, and round-tripping capability. In the following, we describe each category in more detail.

2.1 Multi-Platform

An augmented reality application can be used across heterogeneous computing platforms spanning over head-mounted display devices to mobile hand held devices. Each computing platform can have different properties regarding hardware and sensor, operating system, used AR SDKs etc. To support multi-platform AR experience across heterogeneous computing platforms, an efficient way of developing various AR applications is needed.

2.2 Adaptation

For supporting context-aware and adaptive AR applications various aspects have to be taken into account.

First of all, context monitoring is an important prerequisite for enabling context-aware applications in general. An important challenge in this regard is to continuously observe the context-of-use of an AR application through various sensors. The context-of-use can be described through different characteristics regarding user (physical, emotional, preferences etc.), platform (Hololens, Handheld, etc.), and environment (real vs. virtual environmental information). Due to the rich context dimension which is spanning over the real world and virtual objects, it is a complex task to track and relate the relevant context information to each other. The mixture of real (position, posture, emotion, etc.) and virtual (coordinates, view angle, walk-through, etc.) context information additionally increases the aspect of context management compared to classical context-aware applications like in the web or mobile context.

Based on the collected context information, a decision making process is required to analyze and decide whether conditions and constraints are fulfilled to trigger specific adaptation operations on the AR application. In general, an important challenge is to cope with conflicting adaptation rules which aim at different adaptation goals. This problem is even more emphasized in the case of AR applications as we need to ensure a consistent display between the real world entities and virtual overlay information. For the decision making step it is also important to decide about a reasoning technique like rule-based or learning-based to provide a performant and scalable solution.

As AR applications consist of a complex structure and composition, an extremely high number of various adaptations is possible. The adaptations should cover text, symbols, 2D images and videos, as well as 3D models and animations. In this regard, many adaptation combinations and modality changes increase the complexity of the adaptation process.

2.3 Round-trip

Beside the before mentioned challenges, it is important for a context-aware AR application to support the flexible usage of various information objects. On the one hand information objects can be text, symbols, 2D and 3D objects which are predefined and available in an existing object repository. On the other hand, it should be also possible to digitize existing real world physical objects, e.g. through a 3D scan, so that further objects can be stored in the object repository and reused at runtime. We call this flexible way of transferring real world physical objects in to a repository and making them reusable again as round-trip.

3 Solution Idea

In order to support the development of context-aware augmented reality applications, we have identified basic architectural patterns to address the identified challenges: Multi-platform, Adaptation and Round-trip capabilities.

3.1 Multi-platform capability

For increasing the efficiency of multi-platform user interface development in the context of AR, we envision to establish a model-based development process. Based on the *CAMELEON Reference Framework* [2], as described in Figure 1, we propose a stepwise model-based development process.

The top layer Task & Concepts includes a task model that is used for the hierarchical description of the activities and actions of individual users of the AR user interface. The abstract user interface (AUI) is described in the form of a dialogue model that specifies the user's interaction with the user interface

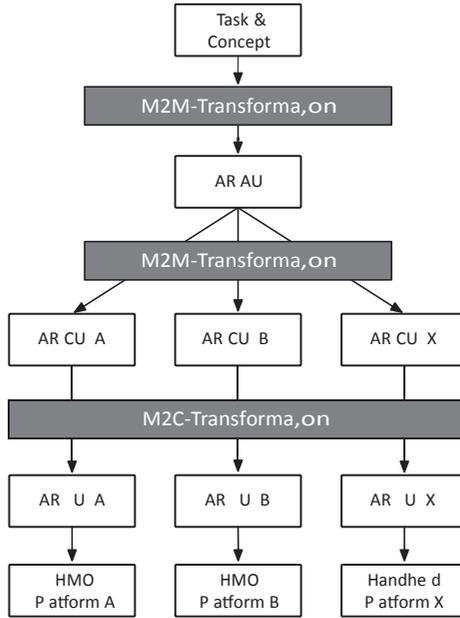


Fig. 1: Multi-platform support.

independent of specific technology. The platform specific representation of the user interface is described by the concrete user interface (CUI), which is specified by a presentation model. The lowest layer of the framework is the final user interface (FUI) for the target platform. The vertical dimension describes the path from abstract to concrete models. Here, a top-down approach is followed, in which the abstract description of relevant information about the user interface (AUI) is enriched to more sophisticated models (CUI) through model-to-model transformations (M2M). Subsequently, the refined models are transformed (model-to-code transformation, M2C) to produce the final augmented reality user interface (AR FUI). Based on this architectural pattern, it is possible to enable multi-platform capability for the different UIs that are generated during the development process.

3.2 Adaptation capability

Based on our previous work in the area of UI adaptation for web and mobile apps [7], we propose an extended version of IBM’s MAPE-K architecture (shown in Figure 2) to support context-aware AR applications.

As depicted in Figure 2, the MAPE-K architecture consists of two main parts *Adaptation Manager* and *Managed Element*. In our case, the *Managed Element* is an AR application consisting of *Tasks*, *Scenes* and *Interrelations* between

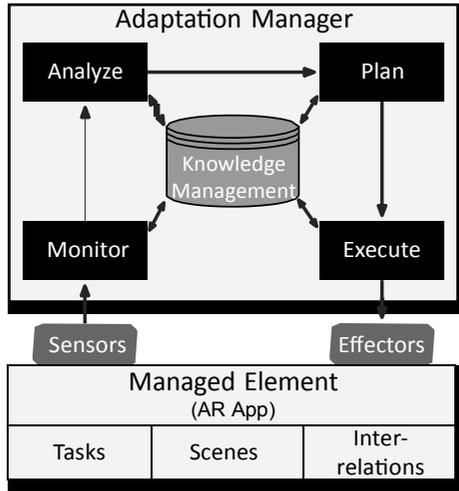


Fig. 2. Adaptation support.

them. The *Adaptation Manager* is responsible for monitoring and adapting the AR application through sensors and effectors in order to provide a highly usable AR experience. In the following, the functionality of each subcomponent of the *Adaptation Manager* is briefly described.

The monitor component is responsible for observing the context information. Context information changes are then evaluated by the analyze component to decide whether adaptation is needed. If so, the planning of an adaptation schedule is done by the plan component. Finally, the adaptation operations are performed by the execute component, so that an adapted UI can be presented. The knowledge management base is responsible for storing data that is logged over time and can be used for inferring future adaptation operations.

3.3 Round-trip capability

For supporting roundtrip functionality in a context-aware AR application, we envision to establish a client-server architecture that enables digitization, storage and reuse of physical objects in an object repository. For this purpose, as depicted in Figure 3, we propose a *AR/VR Server* consisting of an *AR/VR Object Repository*. This repository can contain already predefined virtual objects. On the other hand it is possible to use the *AR Client*, e.g. a handheld AR device, to scan and digitize physical real world objects. These objects can be refined and added to the local *AR/VR repository* which is synchronized with the central *AR/VR Object Repository*. This enables the user to transfer physical objects into the repository, in order to build an object basis as well as projects the repository objects back into reality via augmentation.

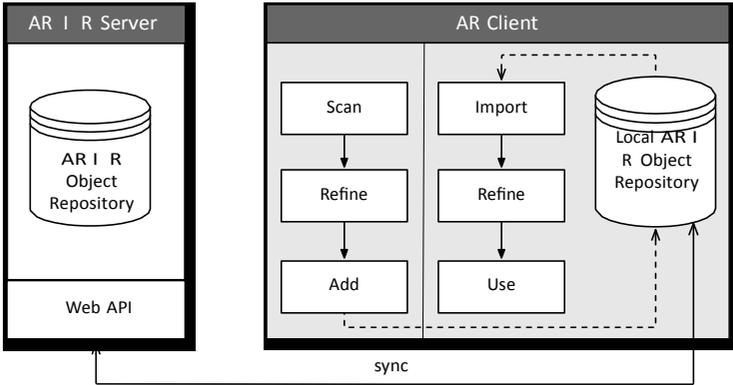


Fig. 3. Round-trip support.

4 Model-based Framework for Context-aware AR Application

In the previous section, we have presented different architectural patterns for supporting the development of context-aware AR applications. While these patterns address basic solution concepts for tackling the different challenges, it is important to design an integrated framework which combines the several aspects of multi-platform capability, adaptation capability and roundtrip capability. For this reason, we propose an integrated model-based framework for context-aware AR applications. Our framework is depicted in Figure 4 and consists of the previously described solution patterns. At design time, the described model-based development process supports to generate the final AR user interfaces for various target platforms. The generated final UI is deployed to a specific AR client which enables the described roundtrip functionality at runtime. Also, the generated final UI of the AR application is monitored and adapted through the *Adaptation Manager* at runtime as described in the previous section.

In addition to the provided framework, we elaborate on the adaptation process as it is a crucial prerequisite for enabling context-aware AR applications.

To address the adaptation process at different development stages, we combine our previous work on model-driven development of adaptive UIs for web and mobile apps [7] with an existing method for structured design of AR UIs [6]. As shown in Figure 5, our solution concept addresses three different aspects: *AR UI*, *Context*, and *Adaptation*. Regarding the *AR UI* aspect, shown in the leftmost column in Figure 5, we rely on the approach and the SSIML/AR language of Vitzhum [6]. SSIML/AR (*Scene Structure and Integration Modeling/ Augmented Reality*) is a visual modeling language which provides model elements for modeling virtual objects and groups in a virtual scene. Additionally, the relations between application classes and the 3D scene can also be specified. Using SSIML/AR, an abstract specification of the user interface of the AR application is created. This *Abstract AR UI Model* is the input for the *AR UI*

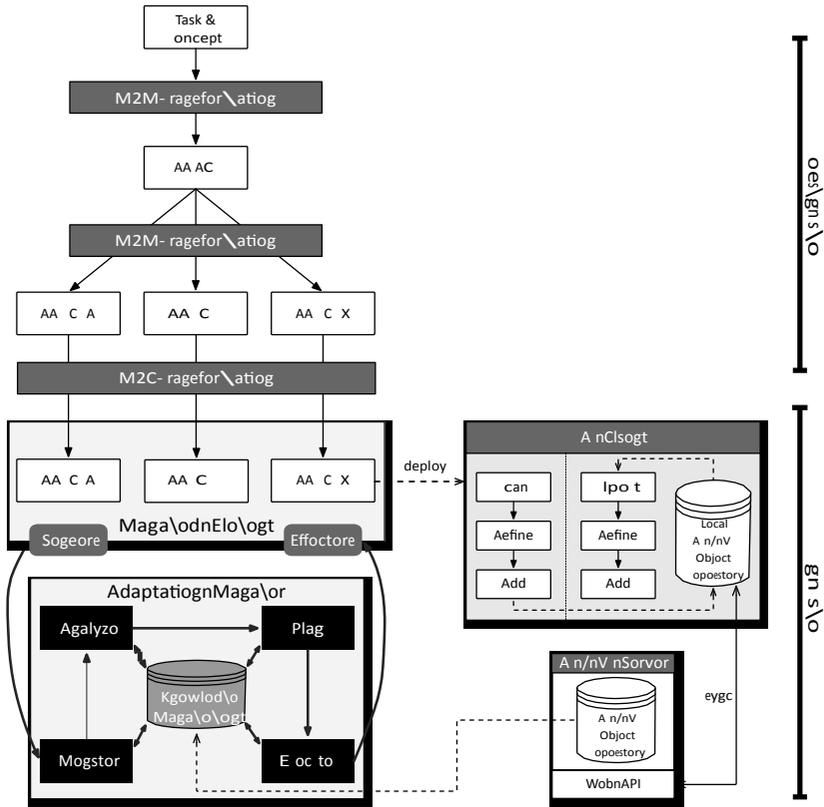


Fig. 4. Model-based Development Framework for Context-aware AR Apps.

Generator, which generates the *Final AR UI*. In order to support the creation of contextaware AR apps, we complement the development method with two additional aspects, namely the *Context* and *Adaptation*, originally presented in [7]. The *Context* aspect serves to characterize the dynamically changing context-of-use parameters by providing an abstract specification in terms of a *Context Model*. Based on the *Context Model*, the *Context Service Generator* generates the *Context Service* which monitors context information like brightness, acceleration or noise level. The *Adaptation* aspect addresses the specification of the adaptation logic in terms of abstract AR UI adaptation rules represented as the *Adaptation Model*. The specified AR UI adaptation rules reference the *Context Model* to define the context constraints for triggering adaptation rules and they also reference the *Abstract AR UI Model* to define which AR UI elements are scope of a UI adaptation change. The *Adaptation Model* is the input for the *Adaptation Service Generator* which generates an *Adaptation Service*. At runtime, the *Adaptation Service* monitors the context information provided by the *Context Service* and adapts the *Final AR UI*.

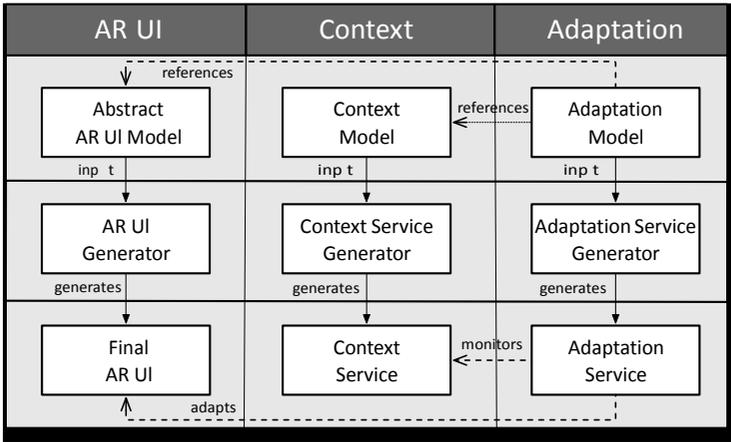


Fig. 5. Model-based Solution Architecture for Adaptive AR Apps.

5 Conclusion and Outlook

This paper discusses main challenges in developing context-aware augmented reality applications and presents architectural solution patterns to address them. Based on the identified architectural solution patterns, we propose an integrated model-based development framework for context-aware AR applications. Furthermore, we elaborate on the adaptation process and propose a model-based solution architecture for adaptive AR applications.

In future work, we plan to implement tool-support for model-based development of context-aware AR applications. Our goal is to support the efficient development of context-aware AR applications for different application scenarios from various domains.

References

1. Azuma, R.: A survey of augmented reality. *Presence* **6**(4), 355–385 (1997)
2. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A unifying reference framework for multi-target user interfaces. *Interacting with Computers* **15**(3), 289–308 (2003)
3. Caudell, T.P., Mizell, D.W.: Augmented reality: an application of heads-up display technology to manual manufacturing processes. In: *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*. vol. ii, pp. 659–669 vol.2 (Jan 1992)
4. Grubert, J., et al.: Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE Trans. Vis. Comput. Graph.* **23**(6), 1706–1724 (2017). <https://doi.org/10.1109/TVCG.2016.2543720>, <https://doi.org/10.1109/TVCG.2016.2543720>

5. Sutherland, I.E.: A head-mounted three dimensional display. In: Proceedings of the December 9–11, 1968, Fall Joint Computer Conference, Part I. pp. 757–764. AFIPS '68 (Fall, part I), ACM, New York, NY, USA (1968)
6. Vitzthum, A.: SSIML/AR: A visual language for the abstract specification of augmented reality user interfaces. In: IEEE Symposium on 3D User Interfaces. pp. 135–142 (2006)
7. Yigitbas, E., et al.: Self-adaptive UIs: Integrated model-driven development of UIs and their adaptations. In: Proc. of the ECMFA 2017. pp. 126–141 (2017)
8. Zhu, J., et al.: A context-aware augmented reality assisted maintenance system. *Int. J. Computer Integrated Manufacturing* **28**(2), 213–225 (2015)