



# On the Development of Context-Aware Augmented Reality Applications

Enes Yigitbas<sup>(✉)</sup>, Ivan Jovanovikj, Stefan Sauer, and Gregor Engels

Paderborn University, Fürstenallee 11, 33102 Paderborn, Germany  
{enes.yigitbas,ivan.jovanovikj,stefan.sauer,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. The benefit and applicability of our solution idea is shown on the basis of two example mobile AR applications supporting context-awareness.

**Keywords:** Augmented reality · Context-awareness · Model-based development

## 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 Sutherland's head-mounted three dimensional display "The sword of Damocles" [10] 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 HoloLens<sup>1</sup> or the Magic Leap One<sup>2</sup>:

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

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

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 (inter-relations between real and virtual information objects) of AR applications [11], 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 [16], a systematic method for supporting the efficient development of context-aware AR applications is not fully covered yet. Therefore, 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 Sect. 2, we discuss main challenges in developing context-aware AR applications. In Sect. 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. In Sect. 5, we present case studies to show the applicability of our framework. Section 6 gives an overview of related work. Finally, Sect. 7 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 (HMD) devices to mobile hand

held devices. There are a number of AR capable devices on the market which are developed by different companies and organizations. Target devices could be smartphones, tablets or HMDs as the Microsoft HoloLens, RealWear Glasses or Google Glass Enterprise Edition, just to mention a few. The cost of a device, its comfort in using it and the ability of a device to help the user accomplish her task are some of the reasons that influences the type of equipment that different users and organizations use to acquire them. In a real world scenario, most users and organizations would like to have the freedom of choosing devices for task execution on an AR application which are most suitable for them if they require to use AR solutions. Mostly, their concerns are to have a mixture of several devices to suit the varying nature of tasks available and also to meet their budget constraints. 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 run-time. We call this flexible way of transferring real world physical objects in to a repository and making them reusable again as round-trip. In order to support such a round-trip for the development of context-aware AR applications, a solution should enable the following steps: scan physical objects, refine object model, store object model in repository. Beside that, the import and usage of the created object models the development environment should be supported.

## 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 Fig. 1, we propose a step wise model-based development process.

The top layer Task & Concept 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 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.

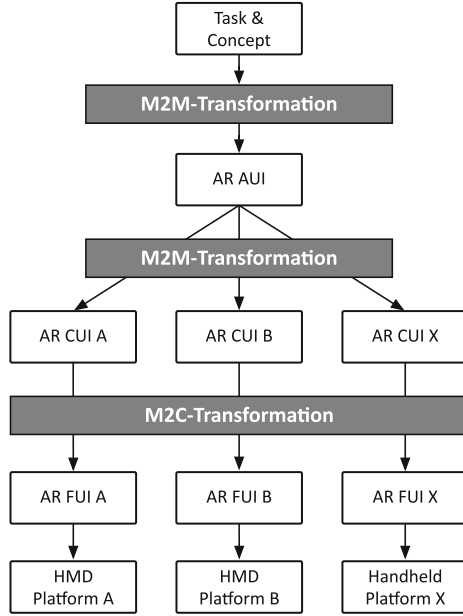


Fig. 1. Multi-platform support

### 3.2 Adaptation Capability

Based on our previous work in the area of UI adaptation for web and mobile apps [12], we propose an extended version of IBM's MAPE-K architecture (shown in Fig. 2) to support context-aware AR applications.

As depicted in Fig. 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 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 sub-component 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.

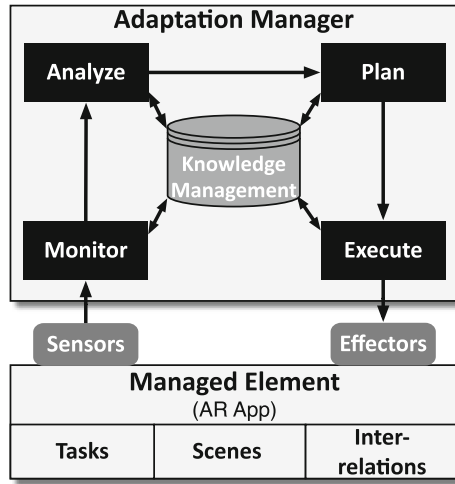


Fig. 2. Adaptation support

### 3.3 Round-Trip Capability

For supporting round-trip 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 Fig. 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 add 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.

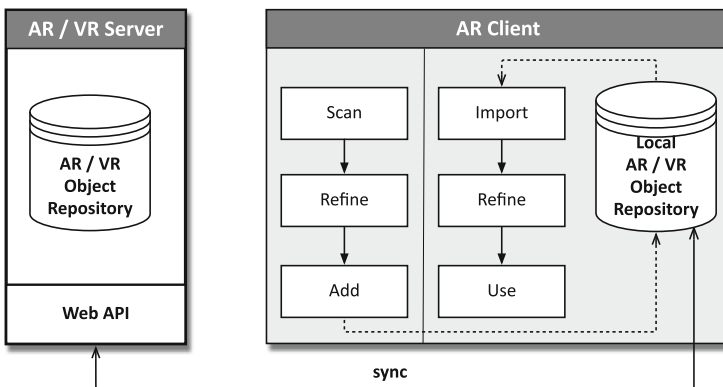


Fig. 3. Round-trip support

## 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 round-trip capability. For this reason, we propose an integrated model-based framework for context-aware AR applications. Our framework is depicted in Fig. 4 and consists of the previously described solution patterns.

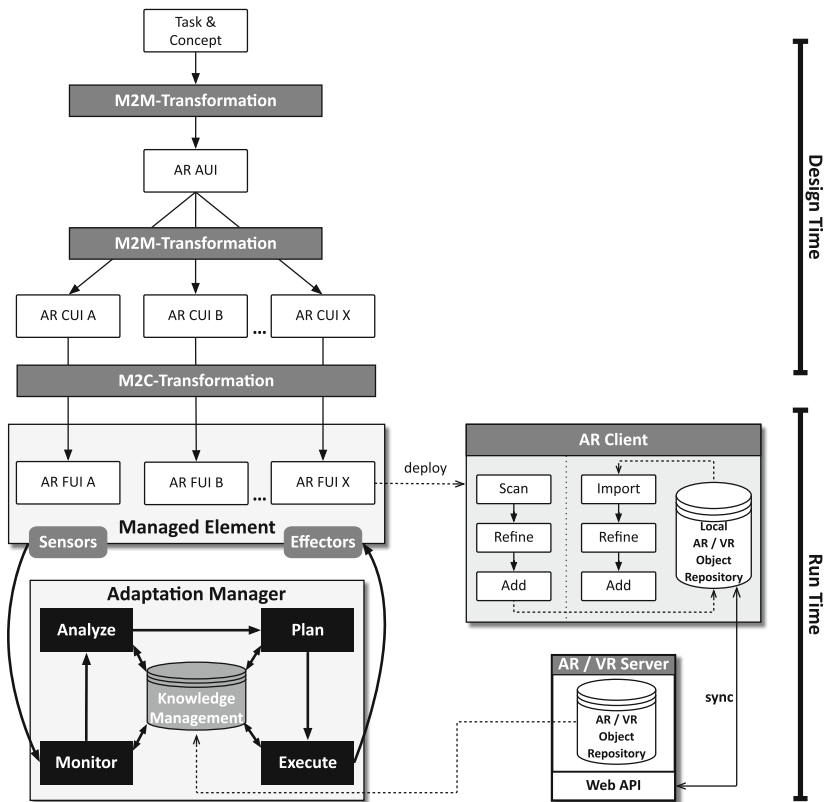
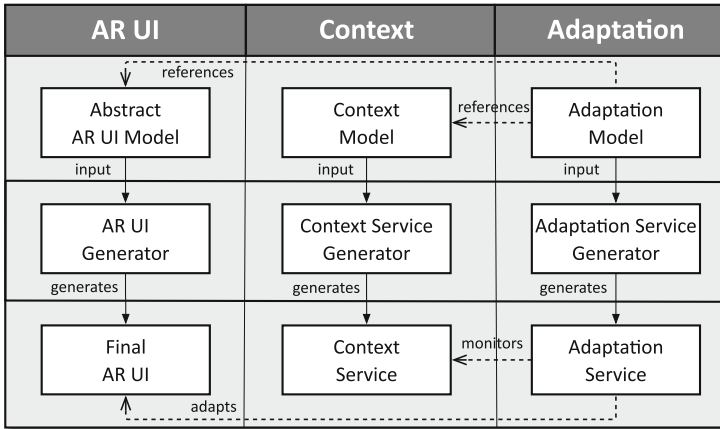


Fig. 4. Model-based development framework for context-aware AR apps

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 round-trip functionality at run-time. 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 [12] with an existing method for structured design of AR UIs [11]. As shown in Fig. 5, our solution concept addresses three different aspects: *AR UI*, *Context*, and *Adaptation*. Regarding the *AR UI* aspect, shown in the leftmost column in Fig. 5, we rely on the approach and the *SSIML/AR* language of Vitzhum [11]. *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 Generator*, which generates the *Final AR UI*. In order to support the creation of context-aware AR apps, we complement the development method with two additional aspects, namely the *Context* and *Adaptation*, originally presented in [12].



**Fig. 5.** Model-based solution architecture for adaptive AR apps

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*.

## 5 Case Studies

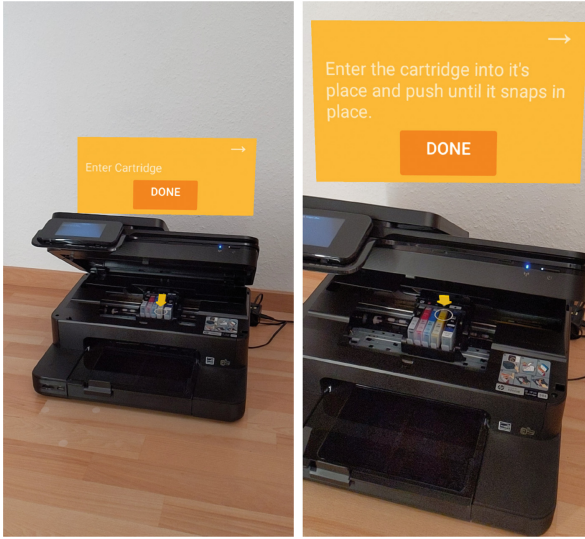
In this section, two case studies are presented which show the benefit of using our model-based development framework for developing context-aware AR applications. The first case study deals with the development of a context-aware mobile AR app which supports UI adaptation capabilities. The second case study shows an example implementation of our object repository supporting the mentioned round-trip capability for scanning and reusing various objects in AR applications.

### 5.1 Context-Aware Mobile AR App Supporting Adaptation Capability

For addressing the adaptation capability, described in Sect. 2, we shortly present the implementation of a context-aware mobile AR application for printer maintenance tasks. Usually, printers have to be maintained and various tasks like refilling paper or changing cartridges should be performed on a regular basis. For such tasks, the printer device usually comes up with a manual. However, most users find it cumbersome to read through the whole manual to accomplish straightforward tasks. Due to this reason and to illustrate the benefit of context-aware AR applications, we implemented an AR application for this example scenario by using our development framework. The context-aware printer maintenance AR (CAPMAR) app makes use of the used smartphone's hardware sensors to observe various context characteristics about the user, platform, and environment. Beside that, we have a preliminary starting questionnaire where newly starting users of the CAPMAR app are asked about their task experience, age etc. Based on those static and dynamically observed context information at run-time, the CAPMAR app is able to trigger AR interfaces adaptations to better suit the context-of-use. Figure 6 is showing a screenshot from the CAPMAR app, where on the left app screen a shorter help message is shown, while on the right screen for a beginner user who is not familiar with the task, a more detailed description of the task is depicted. In summary, the implementation of the CAPMAR app, shows the interface adaptation capability of our model-based development framework for context-aware AR applications.

### 5.2 Object Repository Supporting Round-Trip Capability

In order to address the round-trip capability, described in Sect. 3.3, we have implemented an object repository which supports the digitization of physical



**Fig. 6.** Case study 1: context-aware AR printer maintenance app

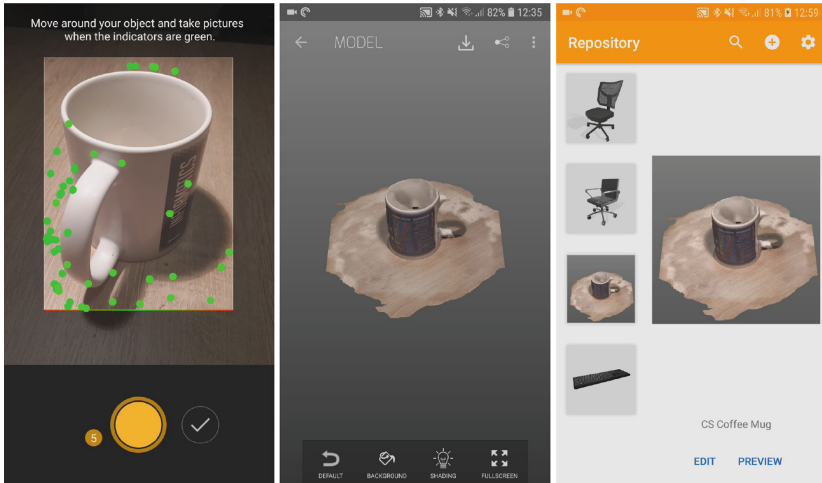
objects through photogrammetry. As illustrated on the left of Fig. 7, we can use third party photogrammetry applications to scan a real object and make a 3D model out of it. The derived 3D object model can be imported into our object repository which enables to manage various of those 3D objects. The object repository is implemented as a web server and supports the persistence of various objects in varying levels of quality detail. Upon the need of the objects, they can be searched and reused based on our mobile client which can position the derived objects in the real scene. In summary, the combination of such an object repository with the concept of context-awareness brings many advantages to ease the developers work in devising context-aware AR apps.

## 6 Related Work

In previous work, different approaches were introduced to address the development of context-aware augmented reality applications. In the following, according the challenges introduced in Sect. 2, we analyze the related work regarding context-awareness in augmented reality applications, multi-platform capability, and object round-tripping.

While context-awareness has been exploited in various types of applications including web [12], mobile [14], cross-channel [15] and virtual reality [13] to improve the usability of an interactive system by adapting its user interface, there are only few existing works focusing on the topic of context-awareness in augmented reality.

In [4], the concept of Pervasive Augmented Reality (PAR) is introduced. A taxonomy for PAR and context-aware AR that classifies context sources and



**Fig. 7.** Case study 2: object repository for AR apps

targets is presented. The context sources are classified as human, environmental and system factors. As apparent in the title, Grubert's work treats Augmented Reality, here with special regards to pervasive Augmented Reality.

Context-aware Mobile Augmented Reality (CAMAR) [9] is an approach on context-awareness in mobile AR focusing on user context, which is measured using the user's mobile device. It enables the user to customize the presentation of virtual content and to share this information with other users selectively, depending on the context. Furthermore, a framework called UCAM (Unified Context-aware Application Model) [5] can be used to create CAMAR-enabled applications. UCAM is a framework which beside the acquisition, process, and awareness of contextual information provides also a unified way of representation with respect to user, content, and environment.

The framework presented in [6] focuses on context-aware adaptation of interfaces in mixed reality, with the main adaptation points being what content is displayed, where it is shown and how much information of it is displayed. It is designed to adjust the content display depending on the user's tasks and their cognitive load and archives this using a combination of rule-based decisions and combinatorial optimization. The framework uses parameters about the applications that are to be displayed as input additionally to the context-specific parameters to achieve a fitting layout optimisation. The framework is mentioning mixed reality as its base, but regarding that it shows contents in the real world and does not create a whole new virtual world, it can safely be said that AR is supported.

While above mentioned approaches address the development of context-aware applications, the main focus is not lying on a model-based development framework for context-aware AR applications. As such, the required concerns and

features regarding context-awareness and interfaces adaptation are mostly manually implemented while our solution idea is aiming a model-based approach that increases reusability, maintainability, and portability of the AR app for different target platforms. The AR Foundation project<sup>3</sup>, aims to the creation of AR apps that run both on iOS and Android. It provides a common API which covers the core functionality of both Android ARCore<sup>4</sup> and iOS ARKit<sup>5</sup>, thus making it possible to develop AR applications for both platforms from a single code base.

Beside the above described approaches focusing on context-awareness in augmented reality applications, we shortly describe prior work that addresses the topics of round-trip capability and object repositories for prototyping and development of context-aware AR applications. One important resource to assist the development of AR applications is given by object repositories such as Google Poly<sup>6</sup> or Unity Asset Store<sup>7</sup>. Main idea of such object repositories is to provide a public repository for all sorts of assets (images, 3D models, sound and visual effects, etc.) which can be reused in the development of augmented reality applications for different domains and in various projects. Beside that, existing approaches like DART [7] or ProtoAR [8] aim to support the rapid prototyping of AR applications. While these approaches ease the prototyping and development of AR applications even for non-technical developers, their focus is not lying on support for addressing context-awareness and interface adaptation. Therefore, our model-based development framework for context-aware AR applications makes use of synergy effects between existing prototyping approaches and object repositories that are combined with a model-based development approach for context-aware AR apps.

## 7 Conclusion and Outlook

This paper discusses the 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 the development of adaptive AR applications. At the end, we show that our framework can be used for the development of context-aware 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 covering different target platforms.

---

<sup>3</sup> <https://unity.com/unity/features/arfoundation>.

<sup>4</sup> <https://developers.google.com/ar>.

<sup>5</sup> <https://developer.apple.com/augmented-reality/>.

<sup>6</sup> <https://poly.google.com/>.

<sup>7</sup> <https://assetstore.unity.com/>.

## References

1. Azuma, R.T.: A survey of augmented reality. Presence: Teleoper. Virtual Environ. **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. Interact. Comput. **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. 2, pp. 659–669, January 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)
5. Hong, D., Shin, C., Oh, S., Woo, W.: A new paradigm for user interaction in ubiquitous computing environment. In: ISUVR 2006, pp. 41–44 (2006)
6. Lindlbauer, D., Feit, A.M., Hilliges, O.: Context-aware online adaptation of mixed reality interfaces. In: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, UIST 2019, New Orleans, LA, USA, 20–23 October 2019, pp. 147–160 (2019)
7. MacIntyre, B., Gandy, M., Bolter, J.D., Dow, S., Hannigan, B.: DART: the designer’s augmented reality toolkit. In: 2003 IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003), 7–10 October 2003, Tokyo, Japan, pp. 329–330 (2003)
8. Nebeling, M., Nebeling, J., Yu, A., Rumble, R.: Protoar: rapid physical-digital prototyping of mobile augmented reality applications. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, 21–26 April 2018, p. 353 (2018)
9. Oh, S., Woo, W., et al.: Camar: context-aware mobile augmented reality in smart space. Proc. IWUVR **9**, 48–51 (2009)
10. Sutherland, I.E.: A head-mounted three dimensional display. In: Proceedings of the Fall Joint Computer Conference, Part I. pp. 757–764, 9–11 December 1968. AFIPS 1968 (Fall, part I). ACM, New York, NY, USA (1968)
11. 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)
12. Yigitbas, E., Stahl, H., Sauer, S., Engels, G.: Self-adaptive UIs: integrated model-driven development of UIs and their adaptations. In: Anjorin, A., Espinoza, H. (eds.) ECMFA 2017. LNCS, vol. 10376, pp. 126–141. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-61482-3\\_8](https://doi.org/10.1007/978-3-319-61482-3_8)
13. Yigitbas, E., Heindörfer, J., Engels, G.: A context-aware virtual reality first aid training application. In: Proceedings of Mensch und Computer 2019, Hamburg, Germany, 8–11 September 2019, pp. 885–888 (2019)
14. Yigitbas, E., Josifovska, K., Jovanovikj, I., Kalinci, F., Anjorin, A., Engels, G.: Component-based development of adaptive user interfaces. In: Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2019, Valencia, Spain, 18–21 June 2019, pp. 13:1–13:7 (2019)

15. Yigitbas, E., Sauer, S.: Engineering context-adaptive UIs for task-continuous cross-channel applications. In: Bogdan, C., et al. (eds.) HCSE/HESD -2016. LNCS, vol. 9856, pp. 281–300. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-44902-9\\_18](https://doi.org/10.1007/978-3-319-44902-9_18)
16. Zhu, J., et al.: A context-aware augmented reality assisted maintenance system. *Int. J. Compu. Integr. Manuf.* **28**(2), 213–225 (2015)