Towards Rapid Digital Prototyping for Augmented Reality Applications

Ingo Börsting
University of Duisburg-Essen
Essen, Germany
ingo.boersting@uni-due.de

Volker Gruhn
University of Duisburg-Essen
Essen, Germany
volker.gruhn@uni-due.de

ABSTRACT

In rapid continuous software development, time- and cost-effective prototyping techniques are beneficial through enabling software designers to quickly explore and evaluate different design concepts. Regarding low-fidelity prototyping for augmented reality (AR) applications, software designers are so far restricted to non-digital prototypes, which enable the visualization of first design concepts, but can be laborious in capturing interactivity. The lack of empirical values and standards for designing user interactions in AR-software leads to a particular need for applying end-user feedback to software refinement. In this paper we present the concept of a tool for rapid digital prototyping for augmented reality applications, enabling software designers to rapidly design augmented reality prototypes, without requiring programming skills. The prototyping tool focuses on modeling multimodal interactions, especially regarding the interaction with physical objects, as well as performing user-based studies to integrate valuable end-user feedback into the refinement of software aspects.

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**;
• **Software and its engineering → Software prototyping**;

KEYWORDS

Augmented reality, prototyping, HCI

ACM Reference Format:
Ingo Börsting and Volker Gruhn. 2018. Towards Rapid Digital Prototyping for Augmented Reality Applications. In *RCoSE'18: RCoSE'18:IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering , May 29, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3194760.3194762

1 INTRODUCTION

Augmented Reality (AR) is used in a wide area of applications, like cultural heritage, education or industry and supports multiple processes by augmenting digital information into the real environment. However, augmented reality is still a novel technology for a majority of end-users.

Until now, the development of AR-applications remains a challenge, since there are no established standards for AR user interface techniques yet. Therefore, there is a necessity of testing users' acceptance of new features within AR-environments. This leads to the assumption, that the inclusion of end-users in prototyping cycles early in the development process can be beneficial, in order to gain valuable feedback, especially for the development of AR-software.

In rapid iterative software development, prototyping allows to iterate through various designs quickly by enabling rapid deployment and feedback loops [6]. Prototyping enables the exploration of design concepts and ideas, helps to refine certain software aspects and empowers the software development team to evaluate design ideas [8]. In AR-software, the realization of appropriate user interfaces and underlying interaction techniques is particularly challenging [3]. Thereby, frameworks for rapid prototyping that allow frequent changes of content, easy adaption of user interface designs and altering interaction designs are needed [10].

Prototypes differ in detail and fidelity, so that the effort developing prototypes can range from low (low-fidelity prototyping) to rather high (high-fidelity prototyping). For short iterations in rapid continuous software development, low-fidelity prototypes are most efficient, because of their fast and cost-effective nature. Until now, low-fidelity prototyping for AR-applications is restricted to non-digital prototyping tools, which allow to conceptualize early design ideas quickly, but can be laborious in precisely capturing interactivity [9]. Regarding novel interaction techniques in AR, designing interactivity is particularly challenging, especially concerning the interaction with physical objects. High-fidelity prototypes are rather time- and cost-intense and require programming skills. Thereby, software designers are usually not able to develop these prototypes [11].

In this paper, we present the concept of a tool for rapid digital prototyping for AR-applications that allows to rapidly develop low-fidelity AR-prototypes, which can be created by software designers, since no programming skills are required. The prototyping tool focuses on user interaction, especially regarding physical objects and gives the opportunity to perform user-based studies to support feedback loops for rapid continuous development.

2 RELATED WORK

2.1 Prototyping

In software development, prototyping helps to determine and validate system requirements and supports the examination of software

RCoSE'18, May 29, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5745-6/18/05…$15.00
https://doi.org/10.1145/3194760.3194762

solutions, as well as the development of user interfaces [13]. De Sá & Churchill [4] describe different use-cases for prototyping in the software development cycle:

- *Probing*, the exploration of new applications, concepts or usages
- *Concept Validation*, the validation of an overall design idea
- *Feature Validation*, the validation of different software-features and -functionalities
- *Usability Testing*, e.g. considering user interfaces
- *User Experience Evaluation*, enables to understand users' feelings and opinions

Prototypes differ in fidelity and effort it takes to build them. It therefore depends on the prototypes' characteristics which prototyping techniques are most efficient for which particular stage of the software development process. In this paper we differentiate between low- and high-fidelity prototypes.

Low-fidelity prototypes are focused on broad underlying design ideas, are produced quickly and aid in the process of generating and evaluating many possible design solutions [2]. These prototypes can be adapted to changing circumstances with low effort and thereby support agility through easy integration of change requests, especially for short iterations. Low-fidelity prototypes are usually produced by software designers and allow to create prototypes on a high abstraction level, so that designers and users without technical expertise can participate in the design process [11]. Nevertheless, these prototypes can be laborious in capturing interactivity [9] and the validation of low-fidelity prototypes is rather difficult [8].

High-fidelity prototypes are usually digital, interactive prototypes with a realistic design, often evolve from low-fidelity prototypes and include realistic visual impression, interaction design, usability and user experience. An advantage of using high-fidelity prototypes is, that users can get a realistic impression of the final system, especially regarding design and interactivity [7]. Nevertheless, high-fidelity prototypes lack in the ability to be updated and adjusted on the fly and their construction is expensive and time consuming [4]. Therefore, high-fidelity prototypes are not efficient for evaluating conceptual approaches [11].

In sum, low-fidelity prototypes are most efficient when used in early stages of software development, especially for short software iterations. High-fidelity prototypes are most efficient in later stages of software development, when the conceptual model is already set.

2.2 User-based studies in software development

By including end-users in the software development process, it is possible to observe and measure users' explicit behavior. Zahedi et al. [14] state, that acquiring end-user input early during the design phase leads to an increasing efficiency, quality and performance. Furthermore, testing a software refinement by integrating user-based studies can also provide data on how to proceed with next iterations [1].

2.3 Challenges for AR software development

While interacting with AR-applications, users can usually not rely on previous experience, due to novel interaction methods. Furthermore, AR-applications include multiple modalities that can be used for interacting with AR-software and may be new to end-users [5]. One of the main challenges for developing software for AR is, that the end-user interacts with virtual models and interfaces, since there is a lack of application developing standards for AR. Therefore, the realization of user interfaces and underlying interaction techniques is a particularly challenging task [5].

3 RAPID DIGITAL PROTOTYPING FOR AUGMENTED REALITY APPLICATIONS

This paper focuses on enabling software designers to rapidly model interactions within augmented reality prototypes, especially regarding interactions with physical objects, so that different design approaches can easily be evaluated and compared in order to optimize software aspects. Furthermore, the question how physical objects can be integrated into AR-prototyping is addressed.

3.1 Integration of physical objects

AR enables end-users to interact with physical objects in a real environment, enhanced by digital data. Until now, the detection of real objects in an AR-application remains a challenge. Object detection using image recognition is a complex procedure that needs to be integrated into systems' code and is error prone, since it may not work for different object shapes or under different light conditions. Therefore, the fast integration of physical objects into AR-prototype construction needs to be examined.

We propose the concept of using interaction frames to define physical objects to interact with. Using this approach, software designers create 3D-frames around real objects and thereby define interaction areas in which the interaction should affect the physical object. Here, software designers can simply frame objects in the real environment using gestural input, without the need for further specifying detailed characteristics, like objects' shape or scale. An exemplary image of a framed physical object is presented in Fig. 1.

When repositioning framed physical objects, the interaction frame would have to be adjusted in order to maintain adequate object recognition. Facing this circumstance, the prototyping tool includes a service for the generation of QR-code markers, which can be exported and attached to physical objects. When scanning the QR-code markers with the AR-device and creating an interaction frame around the real object, the interaction frame will be linked to the physical object. Using this technique, real objects can be repositioned without affecting the object recognition, since the interaction frame is fixed to the physical object. Furthermore, the markers are reusable for multiple physical objects, due to the fact, that the linked interaction frame can be adjusted on the fly.

The presented approach enables software designers to rapidly include physical objects within the real environment and allows to specify interactions affecting physical objects, which can be integrated and updated instantly without requiring programming skills.

3.2 Interaction modeling

The modeling of user interactions in an AR-environment remains a challenging task, due to missing design standards. Allowing for interaction modeling in AR-prototyping, the prototyping tool provides a sequential approach. In this concept, the software designer creates step-by-step interaction sequences. These interaction sequences can be used to define AR-interactions with physical/digital objects that follow a particular sequence of multiple user interactions, similar to storyboard-like approaches.

Further on, an example of a simple prototype interaction, in which a context menu should appear next to a physical object, is described.

After selecting the physical object to interact with, the software designer first defines the gestural input, in this case the pointing gesture *AirTap*, which can either be selected from a preset of gestures or performed directly (sequence 1). In the next step, the designer selects the virtual object that should appear next to the physical object, places it inside the AR-environment and links it to the first sequence (sequence 1.1). This exemplary interaction sequence is presented in Fig. 1. Here, a physical object, in this case a coffee mill, is framed and defined with the identifier *coffee_mill*. After including the gestural input, the virtual object *menu_2* is placed in the AR-environment and linked to the physical object. The coordinates of the virtual object are processed by the prototyping tool and can be adjusted on the fly, by either updating the specific values or repositioning the virtual model within the prototyping tool.

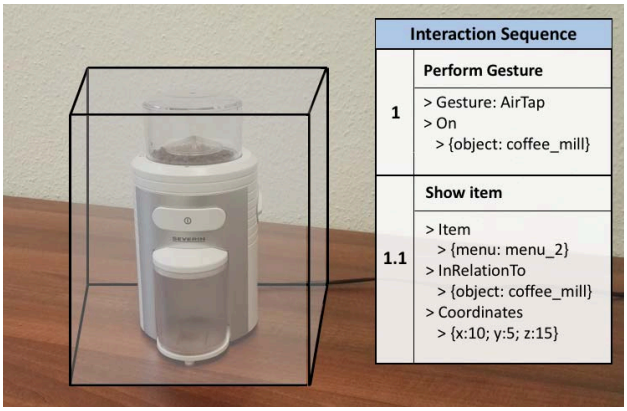

Figure 1: Conceptual image of an interaction frame and interaction sequence for interacting with a physical object

Using this approach, multiple interaction sequences can be applied in parallel. If the context menu should hide by using a voice command, the software designer creates an additional sequence, selects the modality *voice*, creates the voice command by using direct speech and links it to the virtual object (sequence 2). Hiding the virtual object and linking it to the previous sequence will complete this interaction (sequence 2.1).

Every single item of an interaction sequence can be updated, duplicated and adapted to multiple prototyping projects instantly, facing the requirement that a prototyping tool should allow software designers to manipulate prototype components directly and allow easy, rapid modification [2].

Furthermore, the interaction sequence protocol of modeled interactions is exportable and gives an overview of the implemented prototype interactions, similar to a storyboard approach.

3.3 Tool characteristics

Content. In early stages of software design, prototypes usually consist of early content-representations, so that the evaluation of general design principles or interaction methods does not require fully finalized content. In order to support software designers to integrate virtual content, the import of 3D-objects will be supported. Furthermore, external libraries for 3D-models, like Google Poly[1], will be incorporated to enable software designers to select from a preset of available 3D-models.

Multimodality. In order to simplify the modeling of gesture interactions, the prototyping tool will provide a set of predefined gestures that can be used directly to display interactions. This gesture-preset includes gestures supported by most binocular AR-devices, supporting pointing- and selection-gestures, click and drag gestures and manipulation gestures (scale, rotate, tilt). In future work, it is necessary to evaluate if the integration of a full gesture recognition, e.g. by means of a LeapMotion[2]-Controller, is efficient regarding the systems' complexity and the technical skills of software designers.

Voice commands are going to be processed by systems' voice recognition and are reusable for multiple prototyping projects. Different voice recognition libraries have to be evaluated in order to maintain the low-fidelity approach of the conceptualized system.

External input devices, like the *clicker* for the Microsoft Hololens, will be supported in order to simplify or replace gestural inputs, if requested. Types of supported external devices need to be determined, in order to keep the systems' complexity to an adequate level without constraining software designers.

Output. For distribution- and presenting-purposes, a video-file of the AR-prototype interaction can be exported. This gives the opportunity to share AR-prototype experiences with internal or external stakeholders, or within the development team. As described earlier, the interaction sequence protocol of modeled interactions is exportable and may be used as a storyboard-like overview. In Addition, a report containing automatically gathered data, e.g. from user-based studies, can be exported.

User-based studies. In order to gain valuable feedback from user-based studies, usage data will be gathered automatically during the prototype interaction, such as the analysis of users' gaze-behavior. Referring to [12], this data includes:

- *Heat Maps*, which are automatically generated by analyzing users' gaze-behavior
- *Modalities*, tracking of particular used input modalities
- *Interactions*, amount and type of applied interaction methods
- *Duration*, of overall and specific interactions

The automatically gathered data will be combined within a comprehensive report and can be exported to compare different design

[1] https://poly.google.com/
[2] https://www.leapmotion.com/

concepts and to consider insights from user-based studies into software refinement.

4 CONCLUSIONS

We introduced a concept for modeling interactions and integrating physical objects within a digital prototyping tool for AR-applications to enable software designers to quickly construct augmented reality prototypes. Performing user-based studies using the conceptualized tool will support the refinement of software iterations by including end-user feedback.

Furthermore, we presented the idea of enabling software designers to integrate physical objects into AR-prototypes, by introducing interaction frames, which can be created by drawing frames around physical objects. Repositioning physical objects is planned to be supported by attaching QR-markers to real objects.

The modeling of multimodal interactions in AR-prototypes will be supported using the conceptualized interaction sequences approach. Here, software designers are able to model interactions using multiple available modalities in a storyboard-like way, including physical objects and digital content.

The presented approaches will be joined in a comprehensive prototyping tool for AR-applications, allowing software designers without programming skills to create AR-prototypes and iterate through different design concepts quickly. The presented concepts enable the integration of physical objects and the modeling of multimodal interactions, as well as updating software aspects rapidly.

Future Work. In future work, the integration of physical objects using interaction frames will be implemented and evaluated, by realizing a prototypical implementation for the Microsoft Hololens. Nevertheless, the goal of our research is to define a universal approach for implementing AR-prototypes, so that the evaluation results will be adapted to multiple existing binocular AR-systems. The challenge of realizing this approach is on the one hand to enable software designers to create interaction frames within a 3D environment in a user-friendly way and to stick the interaction frame to a fixed position in the environment. On the other hand, the detection of interaction input affecting the interaction frame is a challenge as well. We will initially focus on the technical implementation of the presented approach, especially regarding different types of supported physical objects (shape, size). The concept will be evaluated by a prototypical implementation and conducting a user-based evaluation. The generation of QR-code markers to enable the repositioning of framed objects will be implemented as well, including user-based evaluation to validate the presented approach.

The presented concept of using interaction sequences to enable software designers to model multimodal interactions will be evaluated in future work. The complexity of interactions supported by the presented approach will be validated by conducting user-based evaluation, using a prototypical implementation of the presented concept. This validation will include interactions of diverse complexity to examine the degree of interaction complexity that is going to be supported by the presented approach. Nevertheless, we assume, that interactions can usually be narrowed down to multiple simple interactions, even in complex systems.

Furthermore, several design decisions need to be discussed, such as evaluating voice recognition approaches or supporting full gesture recognition.

Consequently, we aim to adapt software prototyping principles to the realm of software engineering for augmented reality. We strive for enabling the creation of digital prototypes for AR-applications in a time- and cost-effective way, supporting the rapid integration of physical objects, as well as the modeling of multimodal user interactions, in order to optimize the software engineering process of augmented reality applications. Regarding missing design standards for interaction methods in AR-applications, we assume that supporting the integration of user-based studies into the development of AR-applications will increase the effectiveness and usability of future AR-software.

ACKNOWLEDGMENTS

The European Union supported this work through project CPS.HUB NRW, EFRE No. 0-4000-17.

REFERENCES

[1] Jonathan Arnowitz, Michael Arent, and Nevin Berger. 2007. *Effective Prototyping for Software Makers.* Elsevier Ltd, Oxford, Amsterdam ; Boston.

[2] David Benyon. 2010. *Designing interactive systems: a comprehensive guide to HCI and interaction design* (2. ed.). Pearson Education, Harlow [u.a.].

[3] Wolfgang Broll, Irma Lindt, Jan Ohlenburg, Iris Herbst, Michael Wittkamper, and Thomas Novotny. 2005. An Infrastructure for Realizing Custom-Tailored Augmented Reality User Interfaces. *IEEE Transactions on Visualization and Computer Graphics* 11, 6 (Nov. 2005), 722–733. https://doi.org/10.1109/TVCG.2005.90

[4] Marco de Sá and Elizabeth Churchill. 2012. Mobile Augmented Reality: Exploring Design and Prototyping Techniques. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 221–230. https://doi.org/10.1145/2371574.2371608

[5] A. Dünser, R. Grasset, H. Seichter, and M. Billinghurst. 2007. Applying HCI principles to AR systems design. (2007). https://ir.canterbury.ac.nz/handle/10092/2340

[6] Ali Hosseini-Khayat, Teddy Seyed, Chris Burns, and Frank Maurer. 2011. Low-fidelity Prototyping of Gesture-based Applications. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '11)*. ACM, New York, NY, USA, 289–294. https://doi.org/10.1145/1996461.1996538

[7] Jean-Yves Lionel Lawson, Mathieu Coterot, Cyril Carincotte, and Benoît Macq. 2010. Component-based High Fidelity Interactive Prototyping of post-WIMP Interactions. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction (ICMI-MLMI '10)*. ACM, New York, NY, USA, 47:1–47:4. https://doi.org/10.1145/1891903.1891961

[8] Youn-kyung Lim, Apurva Pangam, Subashini Periyasami, and Shweta Aneja. 2006. Comparative Analysis of High- and Low-fidelity Prototypes for More Valid Usability Evaluations of Mobile Devices. In *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles (NordiCHI '06)*. ACM, New York, NY, USA, 291–300. https://doi.org/10.1145/1182475.1182506

[9] Linchuan Liu and Peter Khooshabeh. 2003. Paper or Interactive?: A Study of Prototyping Techniques for Ubiquitous Computing Environments. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 1030–1031. https://doi.org/10.1145/765891.766132

[10] Mathias Müller, Tobias Günther, Dietrich Kammer, Jan Wojdziak, Sebastian Lorenz, and Rainer Groh. 2016. Smart Prototyping - Improving the Evaluation of Design Concepts Using Virtual Reality. In *Virtual, Augmented and Mixed Reality (Lecture Notes in Computer Science)*. Springer, Cham, 47–58. https://doi.org/10.1007/978-3-319-39907-2_5

[11] Jim Rudd, Ken Stern, and Scott Isensee. 1996. Low vs. High-fidelity Prototyping Debate. *interactions* 3, 1 (Jan. 1996), 76–85. https://doi.org/10.1145/223500.223514

[12] Dirk Schart and Nathaly Tschanz. 2017. *Augmented und Mixed Reality für Marketing, Medien und Public Relations* (2 ed.). UVK Verlagsgesellschaft mbH, Konstanz.

[13] Ian Sommerville. 2011. *Software engineering* (9. ed., internat. ed.). Pearson, Boston, [u.a.].

[14] Mithra Zahedi, Guité Manon, and De Giovanni. 2011. Addressing User-Centeredness: Communicating Meaningfully Through Design. In *Designing together: Proceedings of the 14th International conference on Computer Aided Architectural Design (CAAD Futures)*. Les Éditions de l'Université de Liège.