Mixed Reality - A Source for New Authoring Methodologies?

Abawi, Daniel F.
Fraunhofer AGC

Dörner, Ralf
Fraunhofer AGC

Geiger, Christian
FH Harz

Grimm, Paul
Fraunhofer AGC

Haller, Michael
FH Hagenberg

Hartmann, Werner
FAW

Reinhold, Silvan
Fraunhofer AGC

Zauner, Jürgen
FH Hagenberg

1 Motivation

In the past few years, an increasing variety of new input and output devices have become available. And they are not only available in the sense that they exist as prototypes in a research lab, but they can be bought from commercial vendorsat increasingly reasonable prices. This catalyzes the development of new human-computer interaction techniques with a much greater degree of freedom compared with the traditional WIMP (windows, icons, mouse, pointer) interface. As a result, user interfaces have become more sophisticated, technically complex and diverse. They are often adapted to a specific application purpose or even customized to an individual user. This makes the authoring of applications or the creation of content for these applications more complex and difficult to manage. New skills are required of the authors and more, as well as more varied, experts have to participate in the authoring process. Is this the downside of the post WIMP interface, a severe drawback for people that are calling for more reusability, cost efficiency or end-user involvement in the creation of applications? How easy will it be for people to really take advantage of the new developments, to actually use them?

Fortunately, the new input and output devices can also be used for authoring purposes. For example, haptic feedback devices can support an author by providing a new means of constraining the interaction. Mixed Reality, the simultaneous overlay of the real world with a computer-generated world, has a special potential for solving authoring problems. Since the world to be modeled and the actual computer model can be visualized together, it is possible to show their relationship in a less abstract form. This could be very valuable for authoring purposes, since authors are often concerned with building a model of certain aspects of reality.

In this paper, we will offer some thoughts on the question of just how far Mixed Reality (MR) can be used to simplify authoring tasks and to support the author in a new fashion. Further, how can we exploit the potential that Mixed Reality technologies offer us? We will also discuss the related question of what the authoring process of an MR application could look like.

The paper is organized as follows. We start with a discussion of ideas about the authoring process for MR with a focus on authoring tools. Then, we describe the system architecture chosen in AMIRE and show to what extent it is capable of implementing the approach taken. Before summarizing and presenting an outlook on future work, we briefly address some issues concerning the realization of the authoring approach presented.

2 Authoring

2.1 Authoring Roles and Phases

When we talk about authoring in general, we refer to the process of conceiving, creating and customizing an application as a whole or in parts. Keeping this process manageable and efficient despite increasing application size is a difficult task even in traditional application authoring. However, creating Mixed Reality (MR) applications augments the difficulties faced by software engineers even further. In particular, two main problems can be identified that currently complicate the authoring process for MR applications:

1. **Application authoring is mostly reserved for experts** – While writing an application necessitates specific expert knowledge in the particular domain of interest, actually creating or customizing the application itself is not a trivial task either. Domain specificity implies that authoring must take into account high-level information, which can only be provided by a specialist. On the other hand, technical expertise on a lower level is needed to deal with implementation issues, such as object recognition, camera control or tracking. These are equally essential key factors in the production process, and one expert rarely possesses both the technical skills and the application-specific knowledge required to cope with these two tasks single-handedly. In addition, the application domain expert will not want to have to deal with these lower level tasks each and every time they are required.
2. **Authoring an application is difficult even for experts** – Another aspect for the author to take into account when writing applications in Virtual or Mixed Reality is a lack of intuitiveness, regarding, for example, issues of interaction and navigation. Methods of interaction have to provide a mapping between actions in the real world and their corresponding counterparts in the virtual world. Authoring tools must therefore offer intuitive and easy-to-use metaphors and interfaces to the developer, as well as – to a certain extent – to the user with less expertise. In addition, the environment -- or framework -- making these tools accessible must be well-structured and remain flexible at all times, in order to ensure ease-of-use, efficiency and effectiveness.

To address the first issue, it is useful to examine different levels of abstraction in application development and the correlated amount of technical expertise required to deal with the tasks at each of these levels. Since, as mentioned above, it is difficult for one expert to unify domain-specific knowledge and technical expertise, such that both of these aspects can be taken into account in a satisfactory way when authoring an application, it is only natural to break them up and to consider different groups of authors.

This idea should take into consideration that most MR applications today are created in the shape of more or less monolithic blocks with poor re-use capabilities. Porting an application from one domain to another is a tedious, if not impossible task, mostly because many parts have to be re-written from scratch. Therefore, it makes sense to provide building blocks representing domain-specific concepts, which encapsulate the technical aspects of an MR application and provide a high-level interface. These blocks can then be stored in a dedicated repository and consequently be accessed easily for composition and customization to build the application itself.

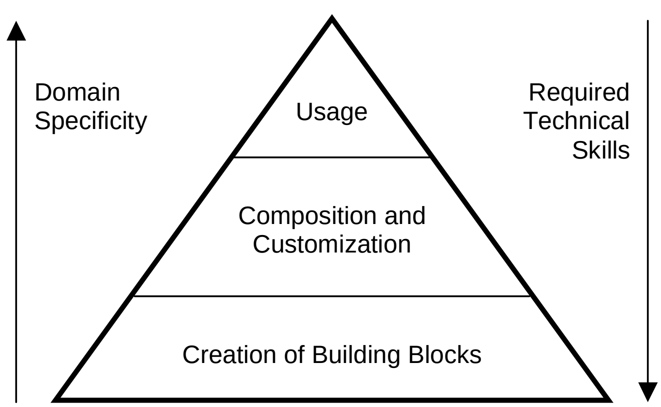

Figure 1: Authoring Groups, Technical Skills and Domain Specificity

A distinction can now be made between the tasks of providing building blocks for applications, composing and customizing them, and using the resulting application. The level of abstraction, or domain specificity, increases from providing building blocks to using an application, whereas the technical skills demanded by each of these tasks decrease (see Figure 1).

Introducing this differentiation allows each of these groups to focus on their own domain of knowledge and expertise, building on the assets provided by the authors on previous levels, in order to provide for a more efficient production process, better re-use ratio, and shorter time-to-market.

Having a repository of customizable building blocks still requires a certain amount of care to be taken during application development. Even though an application now does not have to be completely re-written from scratch -- involving a considerable amount of development time -- the provision of building blocks forces the author to carefully pick those blocks that are needed and useful, and to make them work together as a whole. Focusing on the task of composition and customization, the heart of application authoring with regard to this concept, we can therefore distinguish between different phases of activity [Hain1997]:

- **Qualification** – The core of this phase is the identification of serviceable building blocks. Taking into account the specific needs of the underlying application domain, the usefulness and applicability of building blocks has to be evaluated in the context of thorough requirements engineering.
- **Adaptation** – Individual building blocks are usually created with different application contexts in mind, which means that they have to suit a variety of needs. Therefore, they must be customized to fit the particular requirements posed by other building blocks and by the application itself.
- **Combination** – Placing a set of separate building blocks side-by-side does not yet yield an actual application. Instead, it is necessary to combine the previously qualified and adapted blocks to form a whole, in order to allow for them to interact and unfold their usefulness. This requires a well-defined infrastructure, such as a framework, which will be explained later.

Figure 2 illustrates this process of composition and customization.

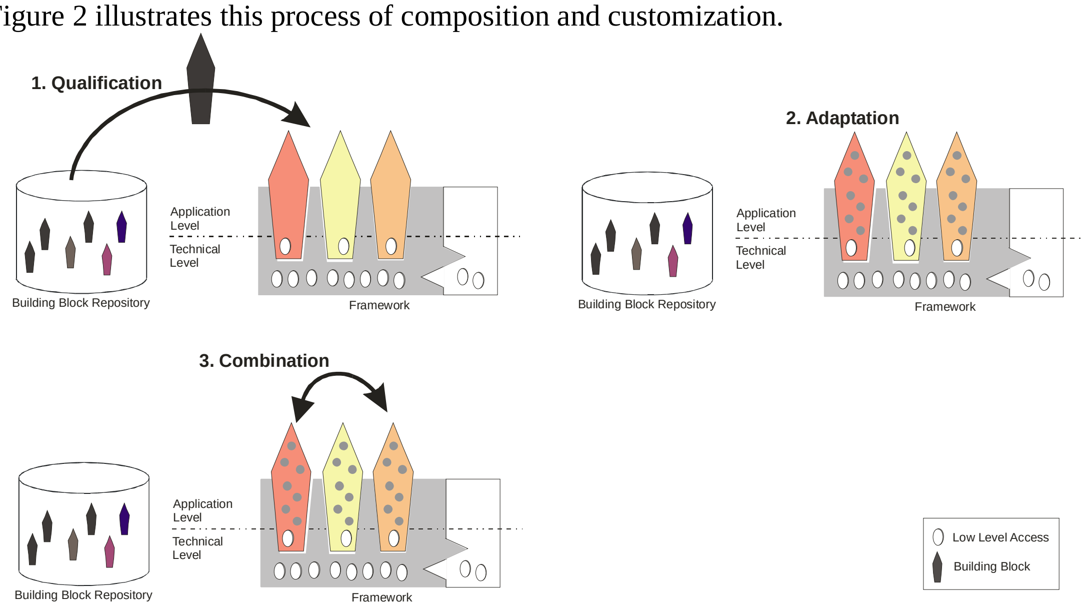

Figure 2: Qualification, Adaptation and Combination of Building Blocks

These tasks should not have to be performed manually, even though building blocks as we describe them are a high-level concept and, therefore, require less technical handling skills by nature. Even so, among other things, customization and the previously mentioned lack of

intuitiveness with regard to interaction and navigation require dedicated authoring tools that ensure ease-of-use not only for users, but also especially for application authors.
Since we defined the distinction between the three different phases during MR application creation and usage, we are able to use this classification to survey the possibilities of enhancing the authoring process. In our opinion, it is illusory and impossibleto provide the user/author, who does not have the necessary knowledge and technical expertise, with tools for creating sufficient and proper building blocks for a task that supposes such technical skills..
So, in this early stage of the authoring process, the process itself is not open to the user with exclusive domain-specific know-how, unless he is also an expert in technical issues. However, the possibilities of component-based content creation show a way out of this problem, since components are solutions for this specific technical layer and for a specific range of use. Nevertheless, authors that resort to components have to have a certain kind of technical knowledge to deal with components. For instance, they have to evaluate the technical qualification of components and their interplay if combinations of several components are to constitute an MR application. All in all, the amount of mandatory knowledge is not comparable to the expertise that is necessary for component development. For the author without technical expertise, this approach encourages a technical abstraction and relocates the necessary knowledge into domain-specific issues. Since the conventional approaches for authoring concentrate primarily on technological features, we require a shifting into a more user-centered approach. To achieve this goal, it is not enough to follow up these appendages through the development of the framework; it is all the more mandatory to realize this paradigm in the several different authoring tools.

2.2 Authoring Tools

The authoring tools assume the prominent role in the authoring creation process, since the author has to interact with the interfaces of these tools to build his application. When we talk about a user-centered approach, we mean the concentration on tools that abstracted from technical issues and fortification the provision of tools with intuitive metaphors. In reference to the classification above, we call for tools and the corresponding metaphors on the domain-specific layer.
This requirement is not surprising as it is not new. In the process of answering the question of innovative tools and metaphors, we have to keep in mind which differences MR technologies offer in comparison to Virtual Reality. VR applications works in the following way. They offer a user interface, which -- in the majority of cases -- is virtual. This means that the user/author has to handle these tools by way of some kind of device (i.e. a mouse).

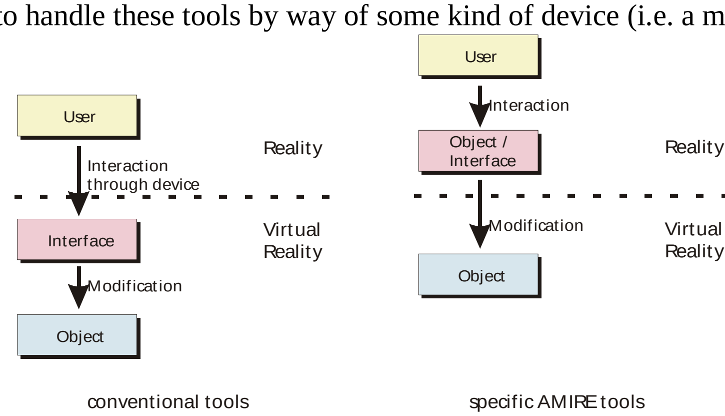

Figure 3: Conventional Tools in Comparison to Specific AMIRE Tools

MR technology distinguishes itself through the fact that a bigger part of the sequence is playing in reality or is at least affected by reality. This gives us the opportunity to open up a new class of authoring tools and metaphors for MR applications. The one thing that these tools have in common is that they do not possess a virtual user interface. These kinds of tools receive their input through modifications of dedicated real objects; therefore, we assert that their interfaces are intuitive for non-expert users also. (see Figure 3)

In the following section, we describe one example of an authoring tool that we want to realize in the project AMIRE. For this case, we have chosen an aspect of 3D applications, where many users/authors encounter difficulties during their daily work – navigation in 3D scenes and the positioning of objects in the virtual scene. Especially for non-experts, these tasks do not seem to be intuitive. This matter predominates if we keep in mind that navigation and positioning are essential tasks within the scope of interaction with VR, as well as in MR. Let us take a look at an author who wants to insert a predefined part during the composition and customization phase. In our example, the predefined component is a glove box, a form of instrument to simulate virtual chemical experimentations in a concluded atmosphere.

The goal of this authoring example is to define the correct position and orientation of the virtual object (in our case the glove box) relative to the real object[1]. For MR authoring, this is an important task, since the size and orientation of the predefined parts are stated by the parts author during the *building blocks* phase (see Figure 4a). The relative position and orientation – which have to be defined – are used during runtime in the usage phase to get an accurate swapping of the virtual and the real object (see Figure 4b and Figure 5b).

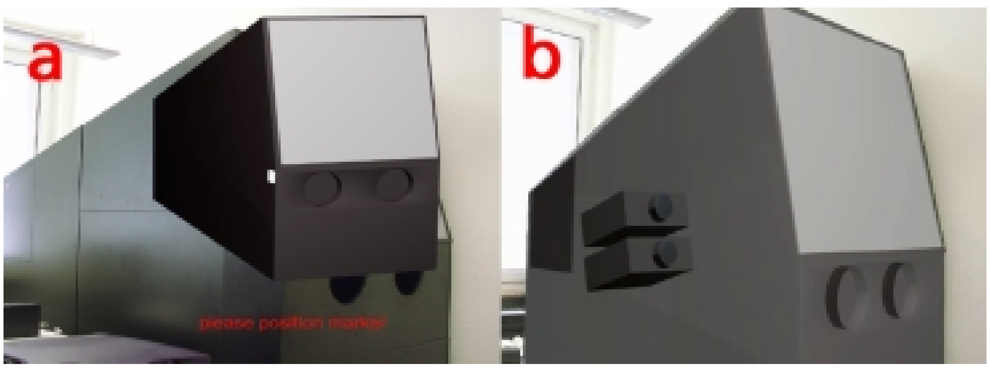

Figure 4: (a) Virtual Object Before Alignment (b) Virtual Object After Alignment

Due to the fact that the author knows very well how to change the position and orientation of real objects in reality, we translate this metaphor – which is intuitive – for our demonstrator in the AMIRE project. Therefore, we establish two specific real objects upon which the user's interaction is based. Due to the position and orientation of these two (real) objects, a position and an orientation for the corresponding virtual object (our glove box) are determined. In contrast to conventional authoring tools, the modification of the orientation and position of the virtual object is done in reality. By increasing the distance between the two special markers, the user/author can modify the size of the virtual object and vice versa. For a modification of the position, the user/author simply needs to move the special markers (see Figure 5a). We act on the assumption that the special markers are attached to a real object, i.e. the glove box. Respective to these metaphors, a change in the orientation of the virtual object is also intuitive. The user has to rotate the special markers or change the tilt – the modification will be converted to according changes of the virtual object. For all of these operations, the system is tracking the position and orientation of at least one of the special markers to affiliate the corresponding parameters for the virtual objects.

[1] To be more accurate: since we are using the ARToolKit, we have to define the position of the virtual object relative to the assigned recognition marker.

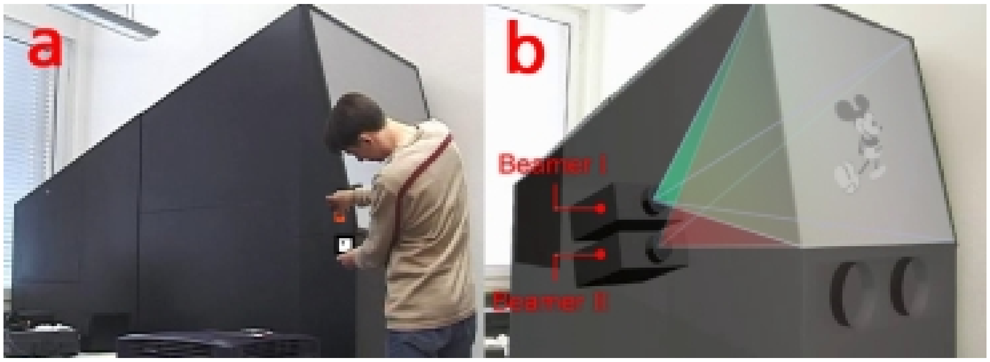

Figure 5: (a) Interaction through Modification of Real Objects; (b) MR Application in Action

3 Architecture

AMIRE's goal and objective is not to develop new base technologies for mixed reality applications, but to adopt existing solutions and provide them in a uniform way. Accordingly, the scientific objectives are focused in the areas of **component-based modeling**, **Mixed Reality** and **human-computer interaction,** especially in authoring environments.

The AMIRE architecture includes the following components:

- MR gems,
- MR components, and
- MR framework

Similar to existing gems collections (game programming gems, graphic gems), AMIRE will produce an **MR gem** collection containing efficient solutions to individual mixed reality problems. A gem could be an object recognition library, a 3ds object loader, a solution for media generation (2D, 3D, audio, text) or a high-level animation. In our case, we concentrate on MR-based gems. The gem collection will be used in the AMIRE framework, but can also be used in other mixed reality projects. It will be ensured that useful, existing MR technology software units will be available in a well-defined way, so that they can be used multiple times in MR applications, as well as in the MR authoring tools. There will be MR gems for using augmented reality technologies (e.g. MR gems for identifying markers in a video image), as well as for using rich media (e.g. MR gems for handling and mapping video in a computer-generated 3D scene). Typically, AMIRE gems can be reused in many different MR-based applications. For example, an object recognition gem can be used to explore a special art piece and to explore the inner parts of a machine. The MR gems in turn can be used to build application-specific **MR components,** as well as an **MR framework** that defines how MR components can communicate with each other and can be integrated in an MR application.

MR components represent solutions for specific problems and they typically combine and extend MR gems towards advanced high-level features of an MR application. MR components feature a unified interface that allows them to be easily configured and combined via the MR authoring tools, using a suitable component model. For example, an object recognition gem, a path animation gem and a 3D object loader gem may be combined into a "magical lens" component that illustrates the inner details of a real machine by providing a suitable animation of virtual machine parts.

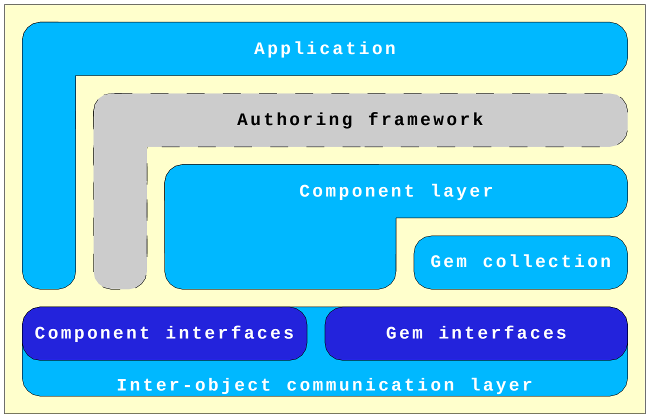

Figure 6: The Framework of the AMIRE Architecture

Figure 6 describes the different layers of the AMIRE architecture. On the base layer, we have the inter-object communication layer, which allows the communication of both the components and the gems. This communication layer provides generic interfaces and mechanisms for component management, communication and authoring. It also includes plugin interfaces for easy gem extension. Further, a gem communication mechanism based on the listener concept is provided. Based on the gem collection, we have the component layer. A component includes one or more different gems. Finally, the authoring tool and the application tool work with all the underlying layers.

The MR framework is the glue of the AMIRE project that integrates both the MR gems and the MR components. The main requirements of the framework are:

- the integration of the Graphics Library (actually, we tend to support OpenGL and/or OpenSG);
- the integration of an object recognition unit (e.g. ARToolkit);
- the integration of different tracking systems (e.g. Polhemus);
- the component interface: this includes a component loader unit that allows the integration of different well-defined objects into the system;
- a message handling architecture, in which the message handler identifies the components that are being addressed, delivers the message and maps it to the corresponding function calls.

4 Realization

The realization of the presented concepts will be accomplished in the scope of the project AMIRE (Authoring Mixed Reality) [Amir2002]. The AMIRE project is funded by the European Commission as part of the action line *Mixed realities and new imaging frontiers*, which focuses on “augmenting reality by enhancing real environments for a wide range of applications, from wearable computing for navigation and engineering processes to content production and interactive entertainment”. The main purpose of AMIRE is to design an authoring environment and a methodology to enable creativity in the creation of advanced mixed reality content.

AMIRE has three main work tasks. First, a best practice example is detailed that shows how MR content can be produced efficiently and how a dedicated authoring application for the user of the MR content can be produced. The approach taken is to identify characteristics of MR methodologies and describe a common infrastructure, employing state-of-the-art software engineering techniques. In addition, the usage of MR technology for authoring tools is examined. Thus, the best practice example consists of the software of two actual demonstrators, as well as generic design recommendations and procedures, interface specifications, dedicated MR authoring metaphors, MR gems and MR components. Second, demonstrators are built that show the benefits of having an authoring tool in addition to the MR application itself. In AMIRE, there will be demonstrators for an oil refinery training application and an application for the Guggenheim Museum. The first demonstrator is built by a group of experts in order to derive an initial version of AMIRE; the second by a consulting company as a kind of experiment in a business scenario setting that will seamlessly lead to an exploitation of AMIRE. Third, evaluation results are derived that show how consultants and content developers can use the AMIRE project results and the economic implications if they choose to offer services in the production of content that makes use of Mixed Reality technologies.

5 Summary and Future Work

In this paper, we presented ideas for new authoring methodologies in the context of the AMIRE project. A production process for MR applications was given and it was divided into three phases: first, a parts creation phase; second, a phase for composition and customization of a MR scene; and, finally, a phase of usage. In a use case description, it was shown that there is a need for innovative tools, especially for the second phase. For these tools, we use user-centered design as a crucial point for a wider acceptance of AMIRE authoring tools. One goal is to enable working directly in MR to create MR content and not to force the author to specify the application on an abstract level only.

Actually, we are working on the first training demonstrator. As future work, we foresee the development of the virtual museum guide as a second demonstrator, as well as the evaluation of the AMIRE tools. After the AMIRE project, we will try to convert the AMIRE results - including authoring tools and methodologies – to the creation of 3D and multimedia creating in general.

6 Acknowledgements

The work described in this paper has been undertaken in the context of the AMIRE project, which consists of the following partners: C-Lab, Guggenheim Museum Bilbao, Fachhochschule Hagenberg, FAW, Fraunhofer-AGC, Labein, OMV, Talent Code and University of Helsinki. The authors wish to thank all of the members of the project team and acknowledge their contributions. AMIRE is partly supported by the European Commission under the IST program (project 34024). Special thanks to Wolfgang Baier and Christoph Bienas for their support.

7 References

ARToolkit ARToolkit
http://www.hitl.washington.edu/research/shared_space

OpenSG Homepage of OpenSG
http://www.opensg.org

[Hain19997] G. Haines, D. Carney, J. Foreman: Component Based Software Development / COTS Integration, Software Technology Review (draft), Revision 97a, Carnegie Mellon University SEL, June 1997

[Schm2001] D. Schmalstieg, M. Billinghurst, I. Poupyrev, H. Katu: *Augmented Reality – The Interface is Everywhere*, Course Notes of Siggraph 2001, Los Angeles, USA, August 2001

[Amir2002] Information about AMIRE
http://www.agc.fhg.de/amire