Efficient Mixed Reality Application Development

Daniel F. Abawi[1], Ralf Dörner[2], Michael Haller[3], Jürgen Zauner[3]

[1]Johann Wolfgang Goethe-Universität, Institute for Computer Graphics, Germany, abawi@gdv.cs.uni-frankfurt.de
[2]Fraunhofer Applications Center, Germany, doerner@agc.fraunhofer.de
[3]Upper Austria University of Applied Sciences, Media Technology and Design, Austria {haller|jzauner}@fh-hagenberg.at

Abstract

In this paper we present the authoring process and the authoring tools for Mixed Reality applications that have been implemented for AMIRE. We introduce concepts that allow non-experts to develop MR applications, with dedicated authoring tools and a well-defined authoring process. The authoring process is reflected in the introduced system architecture. AMIRE is an EU-funded project (IST-2001-34024) where a MR oil-refinery application and a MR museum application are developed.

Keywords: Authoring process, Authoring tool, Mixed Reality, Augmented Reality.

1. Introduction

Nowadays, Mixed Reality applications become more and more popular and thanks to the open source library ARToolKit [1], a lot of new researchers are joining this fantastic area and are developing great applications. Unfortunately, at the moment the development of MR (Mixed Reality) or AR (Augmented Reality) applications can become a very complex task and the researchers have to need very good programming skills to implement their MR programs. The integration of VRML in the ARToolKit environment allows a kind of abstraction in the development. Thus, it allows the development of a complex application with less programming skills. Nevertheless, users have to script their applications and there are no graphical user interfaces that would allow an easy implementation of an MR application with less or no programming skills. Consequently, only programmers or persons with scripting skills are able to implement MR programs. Nowadays, most of the AR/MR applications are implemented, but only few of them are authored by using a tool. In contrast, when we look to the game industry, we will find out that most of the games are implemented and configured with a corresponding authoring tool – mostly with scripting support. The goal of the EU-funded project AMIRE (Authoring Mixed Reality) was to offer an authoring tool for developing MR applications. Thus, concepts are needed to allow developers – who are domain experts outside the field of MR from their profession – to build up their own MR applications efficiently. In our case we had to domain experts. Our first customer comes from the oil industry. In this case the oil-refinery OMV wanted to have a learning and maintenance program based on AR. The second customer comes from the Guggenheim museum of Bilbao, where the visitors should be guided through the museum with AR technology. Such concepts to enable authoring in the field of MR lead to a widespread of the MR technology in general. Thereby, authors with different backgrounds are able to develop MR applications for several application areas on their own. Current AR frameworks like Studierstube, DWARF, and Tinmith-evo5 are mostly object oriented frameworks and/or are based on a component oriented approach, but there is a lack of authoring tools that would support the authors to implement MR applications [2, 3, 4]. Even a scripting support as presented in APRIL would have been too complex for our customers [5].

2. AMIRE Concepts

MR methodologies and technologies are becoming more and more mature, yet they are difficult to apply and to handle by non-experts in this area [6].
Due to the broad applicability of MR methodologies, a broad spectrum of software developers and content authors may want to utilize MR for their individual application area. As a consequence, potential MR authors are not a homogeneous group, but they have significantly different requirements. In particular, this is important for tools that aim to support them in developing MR applications or providing MR content. Our approach is to build an architecture that makes available dedicated authoring tools (e.g. with varied degree of technical abstraction, cf. Figure 1) for each individual author that

feature an individually chosen subset from common functionality.

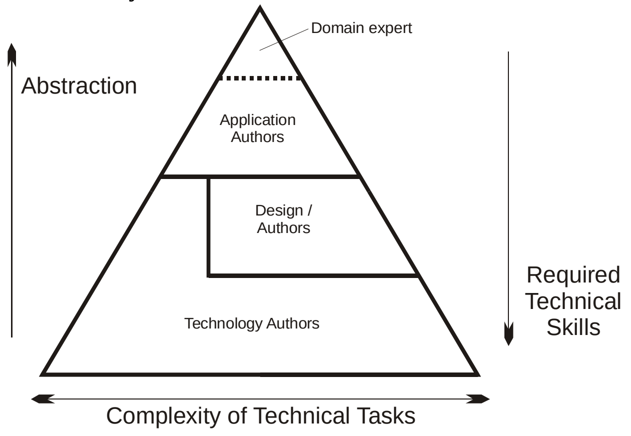

Figure 1. The authoring pyramid, different degree of technical abstraction.

Thus, a major characteristic of a well-designed architecture for MR authoring tools is flexibility in adapting and extending the tools for new classes of authors.
However, just providing user-specific tools still does not guarantee an easy and efficient MR application development, since the process of building the applications of this type is complex. There is no established and well-known procedure how MR applications should be built. Therefore, we conceived a development process that takes the specifics of MR applications into account and hides technical aspects from the author. Our process guides the author step by step through an optimized workflow and ensures that no phases needed are omitted by the author. The main idea in our approach is to let the author assemble an MR application by the reuse (and adaptation) of predefined building blocks (domain-unspecific MR components that can be filled with domain-specific content). The granularity of these blocks turned out to be essential, since the author has to be able to reorganize the scope of his specific MR application into those MR components.
In our development process we identified four major phases:

- qualification of MR components
- adaptation of the MR components selected
- combination of these MR components to unfold their possibility to interact with each other
- calibration of real and virtual objects represented in the MR components

In the qualification phase (cf. Figure 2), the author has to identify MR components that are appropriate with regard to the application objectives. The MR components are provided by a specialist group of MR experts. Taking into account the specific needs of different application domains, the usefulness and applicability of those building blocks have to be evaluated by those experts in the context of thorough requirements engineering.

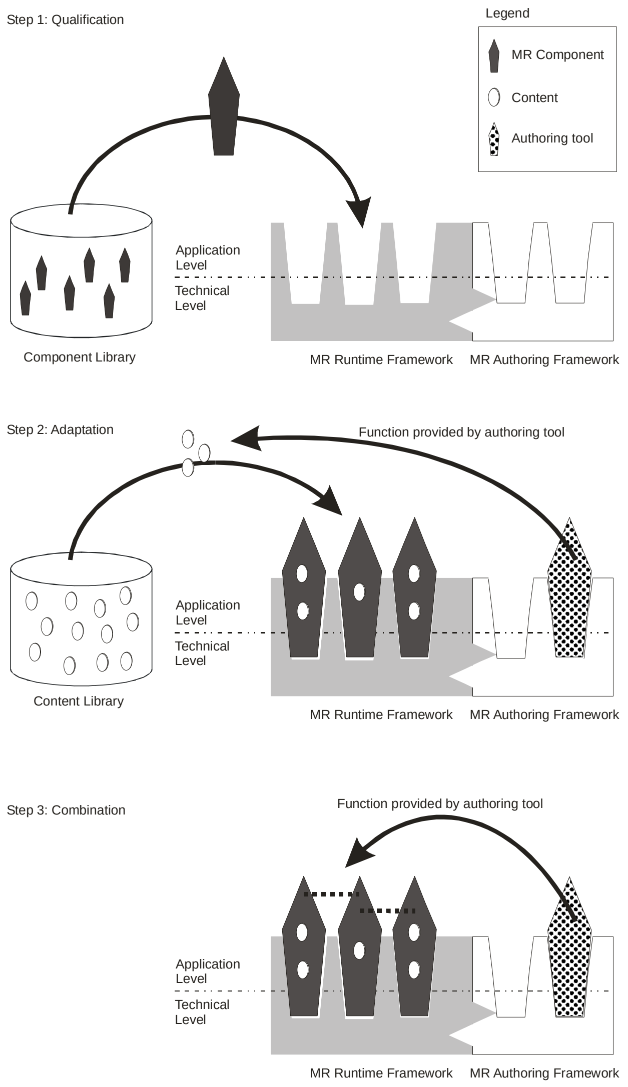

Figure 2: Three phases during the development process.

During the adaptation phase (cf. Figure 2) the author has to customize the MR components to fit the particular requirements posed by other building blocks and by the application itself, since the individual buildings blocks are usually created with different application contexts in mind, which means that they have to suit a variety of needs. For example, a MR component that annotates a real object with 3D text could be customized by defining the color of the text or the font style and size. Application specific content (in our example different texts) can be also made available in the form of a component and may be stored in a central component library. The content components can be used by the author to customize MR components for a specific application.

While the first two phases dealt with MR components individually, in the combination phase (cf. Figure 2) the author defines the relationships and interaction mechanisms between a set of MR components. This task is technically supported due to the fact that all MR components are embedded in a common MR run-time framework and all MR components have according interfaces that allow for the communication with other MR components.

Our development process has to reflect and adequately address the specific requirements of MR. Since in MR the alignment of virtual objects (text, video, annotation, 3d geometries) to reference points in real space is a crucial factor, we introduced an additional fourth phase, the calibration phase. The functionality of tools, that address this phase, aim to calibrate the virtual objects at the correct position (e.g. the author specifies that a virtual character might be aligned to a real chair).

The sequence of the phases in this process is not constrained to be consecutive, thus existing MR applications can be easily maintained and extended by iterating the phases.

Authoring tools can be specifically designed to support the author in accomplishing the tasks in at least one of the four production phases. In our architectural concept, the authoring tool is obtained by coupling an MR authoring framework with the MR run-time framework in which the MR components are embedded. The authoring framework can be customized by specific authoring tool components. These components can be dedicated for different authoring phases. Like the MR components, they can be customized in order to meet the requirements of different authoring groups. Another possibility to meet these requirements is to plug different authoring tool components into the MR authoring framework for authors with different backgrounds. Since there is a well-defined interface between MR run-time framework and MR authoring framework provided in our architecture, the authoring tool components can interact with MR components and thus realize functionality like the qualification of MR components or their customization. One major advantage of our architecture is that it allows the author to adopt the point of view of an end-user. In fact, the MR application is obtained by just decoupling the MR authoring framework from the MR run-time framework. Thus, it can be ensured that the pre-view of the author and the view of end-user can be made identical without any additional efforts.

One particular benefit of being in an MR environment is to use MR technology in order to enhance the usability of MR authoring tools. In fact, in all four phases of our authoring process, MR gives us means to use interaction metaphors for authors that integrate MR technology. Since authoring tasks in MR are inherently space related and MR provides methodologies to interact in space in a direct and intuitive way, this advantage of authoring MR in an MR environment can be exploited by our authoring tools.

Figure 3 shows an example proceeding during the calibration task. An annotation should be inserted at a specific position of the real painting to give the end-user additional information. The author shown in the figure must calibrate the (green) virtual picture frame to precisely match the real picture. The position and orientation of the virtual picture frame changes according to the orientation and position of the reference point on the author's hand. This alignment task is much easier to accomplish for certain groups of authors with an MR metaphor than it is with traditional methods (like measuring calibration data in the real world).

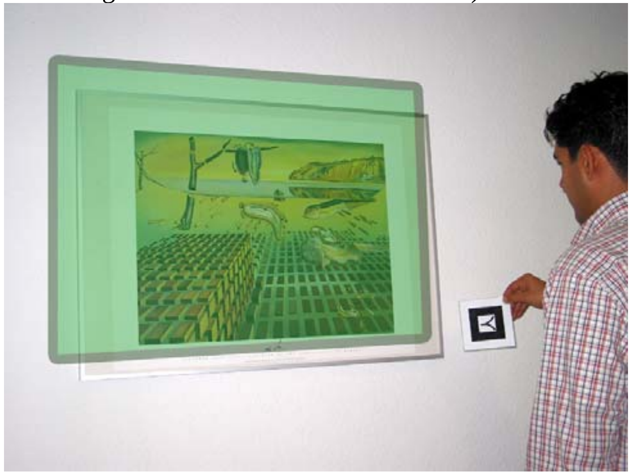

Figure 3: A calibration tool with MR interaction metapher.

With this flexibility in customizing the authoring tool and with the additional degrees of freedom to design the authoring process that are provided by the MR technology itself, MR methodologies can be fruitfully exploited in several application domains. One should note that it is not enough that MR methodologies possess benefits for a particular application domain. The application of MR methodologies must also be feasible for authors of this application domain. This has a direct impact on the costs associated with using MR (another obstacle for a more widespread use of MR) since it is usually more expensive to rely on third party assistance (in this case MR experts) when users wish to create or modify MR applications or MR content.

Our goal to establish MR for several application domains, could be reached even more effectively if the development process is able to be deployed to different authors. AMIRE is open to this approach, during the development of a new MR application; the author is able to enter the process on different technical levels. If the author could not qualify components, he/she is able to develop his own components or even dedicated authoring tools. Less technically orientated is the possibility to integrate content in the predefined building blocks. These steps could be allocated to different authors, as well as the different phases could be allocated to different authors.

The development process should consider that MR is depending on the reference points in real space, therefore the accrued MR applications should easily be maintainable and extendable. Our AMIRE approach reflects this aspect, due referencing the content elements rather than integrate them statically (cf. Figure 2). The constructed MR scene is described in a specific XML-dialect, which includes a description of the needed content elements. These content elements are loaded then during runtime.

Last but not least, we found out that authoring MR applications is easier and more efficient, if the results can be directly sensed without any transformation steps. The AMIRE approach reflects this issue by using (technically) the same software framework to *build and run* the MR application. The authoring framework bases on the technical (run-time) framework and extends this by specific (user specific) authoring tools that encapsulate the functions provided by the technical underlying framework for the author (cf. Figure 2). The framework itself is based on a component-oriented approach. A closer description of the framework can be found in [7, 8].

3. Realization and case studies

We tested the methodology of the authoring process in the oil-refinery environment. The main goal of this demonstrator application was to introduce several stations to new employees of the refinery.

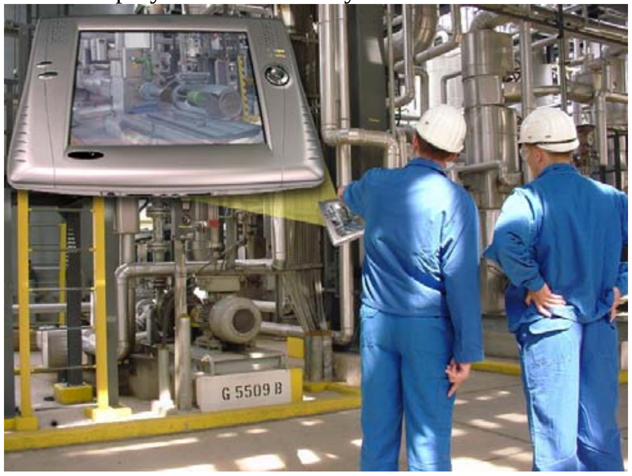

Figure 4: The refinery employees use Tablet PC to get more detailed information of the plant machines.

For the in- and output device we have chosen a Tablet PC because of two reasons: Firstly, it offers provides enough graphic power to render also complex geometry like the inner parts of a pump as shown in Figure 4 and Figure 5. Secondly, it isn't too heavy and too large, and more comfortable than wearing a HMD. Because of safety aspects that have to be considered, the idea of an HMD solution was not suitable for a refinery. The range for the user's field of view is too small and useless for a non toy application.

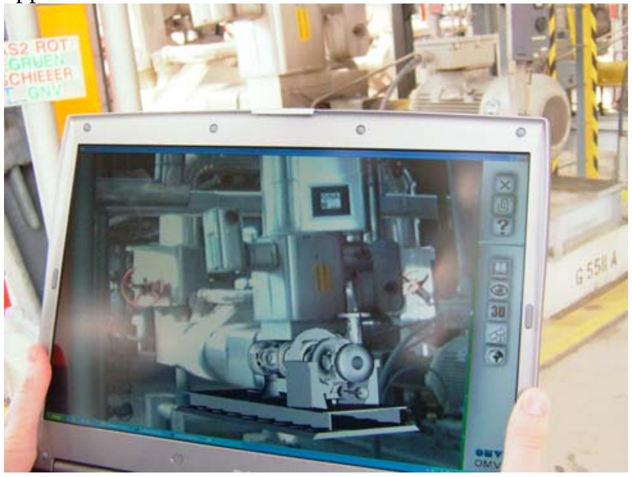

Figure 5: The OMV demonstrator application shows the inner parts of a pump.

As long as the user points with the Tablet PC onto the station a user menu with several interaction possibilities is shown (cf. Figure 6). One of the most important interaction features was the freeze mode of the video image. This freezing metaphor was on the one hand very important for the usability of the application and on the other hand it was also very important for the calibration phase of the authoring process. Due to the jitter problems of the ARToolKit and the bad light conditions in a refinery, a calibration and usage of the oil-refinery application would not have been possible.

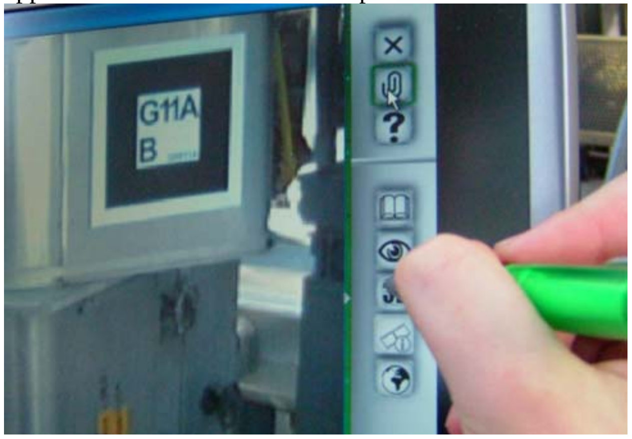

Figure 6: The user interface allows selecting between several important information of the station.

Once the system detected a marker, the user gets more detailed information of the corresponding object (e.g. 3d animations, learning movies, text etc.). Figure 7 depicts for example a schematic overview of a pump.

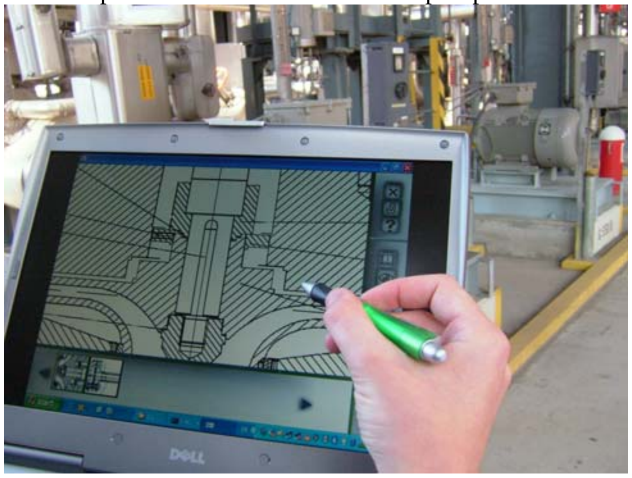

Figure 7: Schematic overview of the pump.

The above described application was implemented with our authoring tool, called connection editor. The editor gives a visual overview of the components (depicted in Figure 8) and it allows manipulating the components and the properties of the components (cf. Figure 9).

After describing the structure of the application by creating the connections between the components and customizing the content like for example the textual descriptions or the geometrical content we had to calibrate the offset between the tracking system and the 3D content. As we have mentioned before, the production process is an iterative process. Therefore, we often had to create new connections and recalibrate the offsets. Without a corresponding authoring tool, this task would be very difficult.

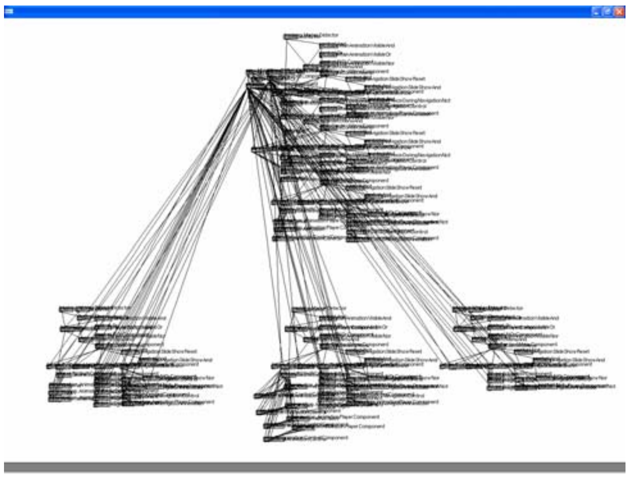

Figure 8: Overview of the OMV demonstrator application as component schema.

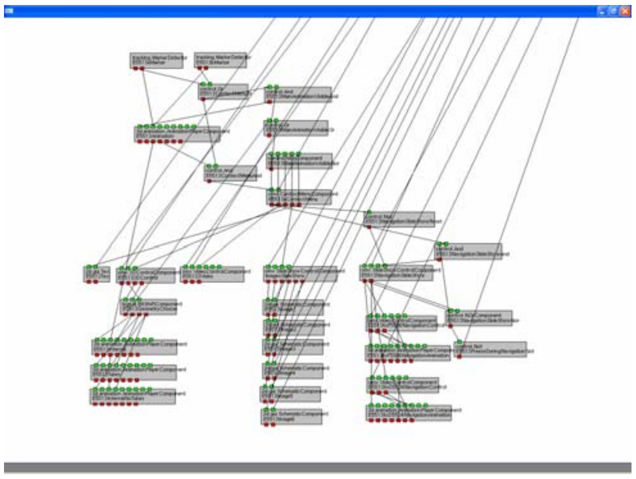

Figure 9: Detailed view of the OMV demonstrator application as component schema.

Figure 8 and Figure 9 show that even in a simple appearing AR application the communication can become very complex. When the author inserts a new component he/she can connect it with the corresponding input and output slots of the components (depicted with the green/red squares). Of course only component slots of the same type can be connected together. By a simple double-click the author can modify the properties of a component. This is one of the most important features of our authoring tool, because it guarantees a high flexibility and reusability of the used components. For example, a 3D component has the property filename of the geometry that has to be modified during the authoring process. The results of the authoring are depicted in Figure 10 and Figure 11.

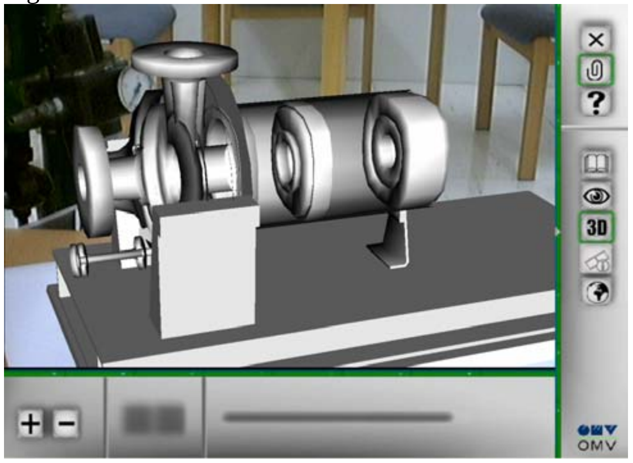

Figure 10: The inside geometry of a pump.

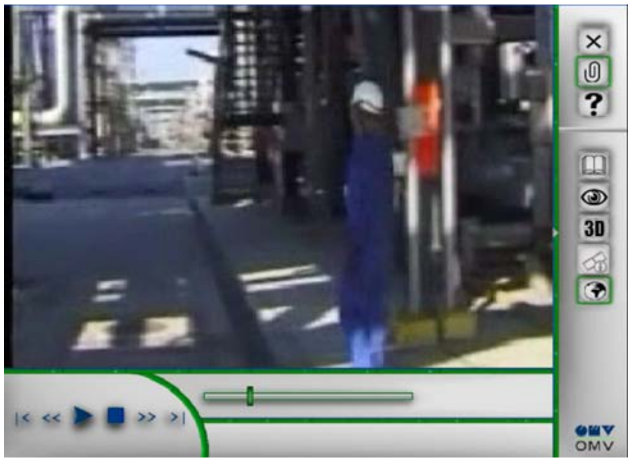

Figure 11: Navigation in a refinery can become very complex. A short movie shows to the employees the right way where they have to go to.

4. Conclusions

In order to achieve a more widespread use of MR technologies we showed that the authoring of MR application as well as MR content by different author groups is a crucial issue. Our approach is to firstly define a development process and secondly conceive authoring tools that are able to support the author in the single phases of the development process as well as to guide the author through this process. Our development process is dedicated to MR development and characterized by four phases (qualification, adaptation, combination, calibration). The authoring process is reflected in a system architecture based on framework and component concepts from software engineering. This architecture can be used to not only implement MR applications according to our development process but also to implement authoring tools that are flexibly adapted to individual authors' needs. As a peculiarity, a direct preview from the end users point of view is provided for the author. In addition, we showed how the authoring tools can be made more intuitive to use by employing MR technology not only for the final application but also for the authoring task itself (e.g. by implementing specific MR-based interaction metaphors for authoring).

Our implementation and application of our concepts in the context of the project AMIRE, e.g. for a MR-based information system in an oil refinery, showed that our development process is flexible to accommodate different end-users needs as well as extensible. Our experience also showed that applications and content created with our development process are far easier to maintain since the authoring phases can be iterated easily – with the support of dedicated authoring tools for maintenance.

5. References

[1] Kato H., Billinghurst M., Blanding B., May R., *ARToolkit*, Technical Report, Hiroshima City University, December, 1999.

[2] http://www.studierstube.org

[3] http://wwwbruegge.in.tum.de/projects/lehrstuhl/twiki/bin/view/DWARF

[4] Piekarski W., Thomas, B. H.: An Object-Oriented Software Architecture for 3D Mixed Reality Application. In ISMAR 2003, The Second International Symposium on Mixed and Augmented Reality, pp. 247-256, IEEE, Tokyo, October 2003.

[5] http://www.studierstube.org/april/

[6] Haller M., Zauner J., Hartmann W., and Luckeneder T., *A generic framework for a training application based on Mixed Reality*, Technical report, Upper Austria University of Applied Sciences, MTD, 2003.

[7] Zauner J., Haller M., Brandl A., Hartmann W., *Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures*, In ISMAR 2003, The Second International Symposium on Mixed and Augmented Reality, pp. 237-246, IEEE, Tokyo, October 2003.

6. Acknowledgements

This research is partly funded by the European Commission under the fifth framework program, project AMIRE (IST-2001-34024). We would like to acknowledge the valuable contribution of all AMIRE project partners.