In: Proceedings of the 11th International Conference on Human-Computer Interaction – HCII 2005, July 22-27, Las Vegas, Nevada, USA.

AMIRE-ES: Authoring Mixed Reality once, run it anywhere

Michael Haller[1], Erwin Stauder[2], Juergen Zauner[1]

[1]Upper Austria University of Applied Sciences
A-4232 Hagenberg, AUSTRIA
[haller|jzauner]@fh-hagenberg.at

[2]Software Competence Center
A-4232 Hagenberg, AUSTRIA
erwin.stauder@scch.at

Abstract

AMIRE (Authoring Mixed Reality), a EU-sponsored research project, is aiming to simplify AR applications so that even ordinary persons, who are familiar neither with computer graphics nor with programming, are enabled to implement their own applications. The component oriented approach of the framework and the flexibility of the system allow an abstraction of the authoring process, thus people can author their application for different systems (e.g. Tablet-PC, PDA,..) with relatively small modifications. In this paper, we present the light version of the AMIRE framework, AMIRE-ES that has been developed for the PDA including a demonstrator example that has been implemented. In this demonstrator, users get additional information during the assembly process of furniture.

1 Introduction and Related Work

Mixed Reality has become a popular buzzword used by many researchers moving from the area of Virtual Reality to Augmented Reality. A lot of time has been spent on improving current tracking systems, finding new ways of interaction, searching for adequate displays, and implementing new exciting applications. But in fact, until now, the *real* killer-application has not emerged (apart from ARQuake developed by Piekarski & Thomas, 2002, where people can kill virtual monsters). A lot of MR-applications are more proof-of-concept studies, but on the other hand it still seems to be really hard to bring this new and fascinating technology to the market.

This was the motivation, when we started in 2000 with the AMIRE[1] (Authoring Mixed Reality) project presented by Abawi, Doerner, Haller, & Zauner, 2004. The main goal of AMIRE was to develop an authoring tool that supports non-MR experts to develop/author MR applications without having programming skills. The idea behind this project was to motivate people (e.g. designers, artists, domain specific experts, etc.) to *author* exciting applications instead of coding them. Thus, the author can concentrate on the content and does not have to think of how to implement the videostream, how to render the 3D geometry or how to integrate the texture into the system. The AMIRE authoring tools are based on a component-based framework, which itself was not implemented on top of CORBA like the projects DWARF (Bauer et al., 2001) and MORGAN (Broll et al., 2004). Our authoring tools include a lot of different tools for the integration of the virtual content, to calibrate it and to define the relationship and the behavior of the different components (cf. Zauner & Haller, 2004). Two demonstrators have been developed to test and evaluate the advantages of AMIRE: a museum demonstrator has been developed for the Guggenheim Museum in Bilbao, where the visitors are guided through the museum using MR technology and a learning demonstrator for an oil refinery company to teach newbies how to control the refinery without having a trainer on the back. For both applications, it was planned to find an easy-to-carry hardware device. It is ironic that a lot of researchers in the field of MR are working with Head Mounted Displays (HMDs), but the usage of such a device is still not accepted by partners from the industry. The reasons are various, but the most important reason is that people do not want to carry an HMD with a heavy laptop in the backpack for rendering the application. The usage of a Tablet-PC is in fact a tradeoff between a device that is light and powerful enough. The Tablet-PC can still be too heavy, even though a Compaq TC-1100 weighs just 1.4 kg. Moreover, after the first tests in the outdoor environment, our refinery partner of the AMIRE project wanted to switch from the Tablet-PC version to a more comfortable device (e.g. PDA or even a mobile phone). This, and the fact that we wanted to explore the flexibility of our framework in the sense of how long it will take to implement a mobile-version of AMIRE, motivated us to implement a light-version of AMIRE with the full power of flexibility regarding the authoring tools. In addition, we

[1] www.amire.net

wanted to explore the performance lags and the rendering losses by implementing a mobile version. Compared to HMDs in combination with personal computers, the authoring of video see-through AR for current PDAs is a really challenging task: low video stream resolutions, little graphics and memory capabilities, as well as slow processors, set technological limitations. Finally, we were interested if the slogan "Authoring Mixed Reality once, run it anywhere" would be just a soap bubble or if it really would bear up. This time, we have to mention rightfully that the interaction metaphors for a Tablet-PC are comparable to those used in a PDA environment; thus, in both applications, users control the MR application by a simple input pen. No different, high-sophisticated interaction metaphors had to be found for both devices, thus making life easier for authors to create their MR application and content and to switch the platform.

In the next section we discuss system architecture of AMIRE and of AMIRE ES, the light-version of the framework. Subsequently, we describe the concepts of the authoring tools and focus especially on CATOMIR, with its visual programming interface approach. The rest of the paper focuses on the implementation and authoring of the application FaiMR (Furniture Assembly Instructor program based on MR technology) that shows a concrete realization of an AR application by using corresponding tools. Finally, we discuss our experiences and results.

2 System Architecture and Implementation

Pintaric, Wagner, Ledermann, & Schmalstieg, 2005 identify three potential candidates for stand-alone handheld AR devices, namely mobile phones, PDAs, and Tablet-PCs. In our environment, we support an authoring environment for PDAs and Tablet-PCs. High end Tablet-PCs have a very good processing power (some of them already support GPU acceleration cards) and a better display compared to PDAs, but they are still too heavy to be carried around and too expensive. The PDA is a good compromise between processing power, size and weight for the mobile setup. Moreover, the user interface of the PDA is widely accepted even by untrained users. On the other hand, the PDA is not the ideal device for 3d graphics rendering, thus a lot of effort has to be invested in optimizing the performance, even though the quality of the display and the processing power has been improved a lot in the last few years. We developed a light version of the current AMIRE framework. Due to the component-based approach, this task was completed relatively quickly (we spent about 4 months for the whole port including the development of new libraries for the PDA platform). Currently, the Pocket PC system supports both Linux and Windows CE. We concentrated on the Windows CE system, though.

A closer description of the AMIRE framework concepts are given in (Haller, Zauner, Hartmann, & Luckeneder, 2003) and in (Zauner & Haller, 2004). It is designed for authoring purpose and AR applications and includes the following main features:

- *A generic configuration mechanism* of components by so-called properties. A property is able to represent data of a base type and structured data. The typing of the structured properties is extendable in an object-oriented way.
- *The communication between components* is based on a slot-approach, where components can exchange data.
- *The framework provides base conventions* for 2D and 3D components such as a 3D and 2D picking mechanism, a user interaction mechanism or placement and alignment
- *Persistence* of an application is supported by an XML based file format. Such an XML file contains a list of library dependencies, the component instances and the connections between them.

Based on the AMIRE framework, we implemented both the authoring tool, and the MR applications which are (in the normal case) authored by the authoring tools (cf. Figure 1). 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 during run-time on the corresponding platform (on the Tablet-PC and on the PDA).

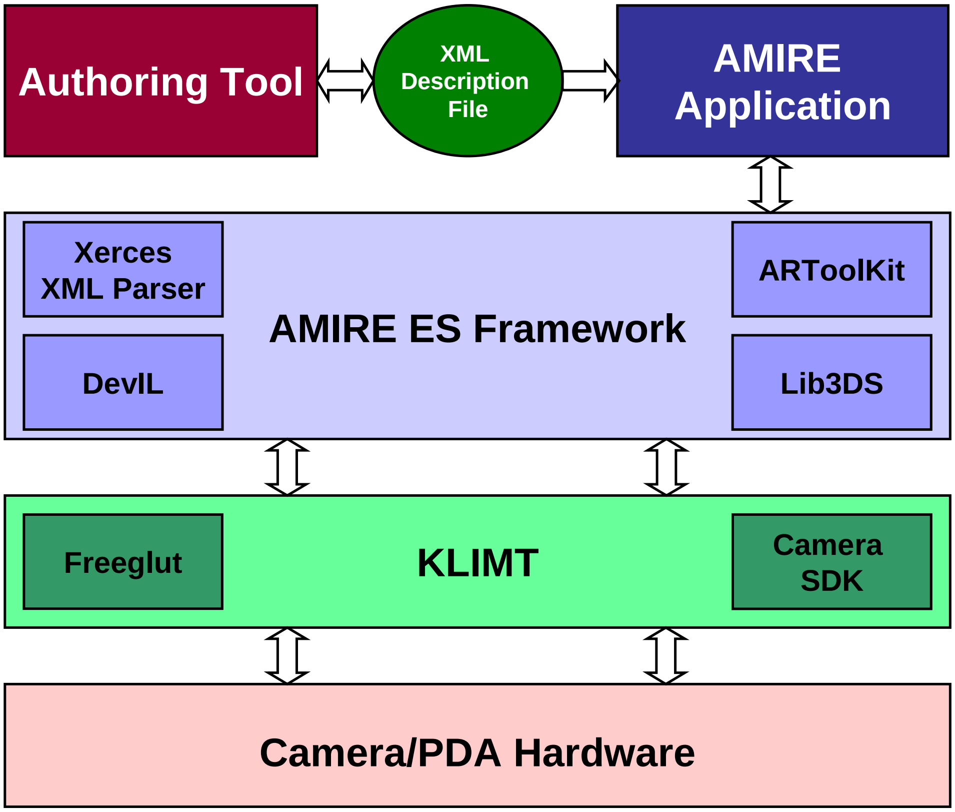


Figure 1: The architecture of AMIRE ES.

Creating the mobile setup for the AMIRE framework was a very challenging task. While the camera image is displayed in the background, 3D graphics is overlaid by rendering it using OpenGL. Most PDAs do not provide built-in camera support. There are several manufacturers for cameras (e.g. Lifeview, HP, Veo, Pretec etc.), but most of them do not offer a corresponding SDK. In our system we used an HP iPAQ 5550 with a CF based Lifeview camera.

On mobile devices such as those of the PDA platform, there is currently no industrial-side support for 3D hardware and software. Although, there have been prior attempts to create libraries similar to OpenGL ES such as PocketGL and TinyGL, these projects are either restricted to 2.5d games or incomplete and discontinued. Our system is based on the Klimt library, developed by Wagner & Schmalstieg, 2003. Klimt is an open-source 3d library, targeted performance and portability with special focus on PDAs. Its API is very similar to that of OpenGL and OpenGL ES. Klimt includes the most important functions of OpenGL and the majority of those of OpenGL ES. Current and upcoming mobile devices do not support floating point in hardware. The lack of a floating point unit on PDAs required a fast fixed-point emulation that avoided divisions.

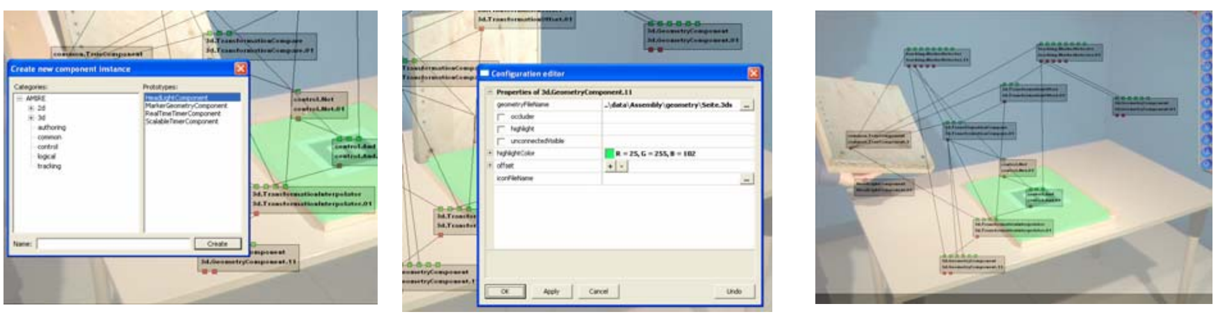
3 Authoring once, run it anywhere

Due to the complexity of different methodologies, developing AR applications can become a really complex task. In addition, it needs a broad spectrum of software developers, graphics experts, and content authors to achieve the goal. Since no single person usually possesses all the expertise required, different author-roles need to be distinguished. Consequently, an authoring tool is required for authors without deep IT- and/or graphics-knowledge. Based on the AMIRE framework, two authoring tools have been developed so far. The first one has been developed by Abawi, Doerner, Haller, & Zauner, 2004. It was mainly designed to be used on a desktop PC. We realized soon that people would love to have a light-version of the authoring tool that is easy to use and powerful enough to author great AR applications by using a Tablet-PC.

(a) (b)

Figure 2: Authoring MR-applications by using a Tablet-PC (a) and the visual programming interface of CATOMIR (b) improves the authoring process dramatically without the requirements of IT and/or graphics.

CATOMIR (Component oriented Authoring Tool for Mixed Reality) is the light version and was firstly introduced by Zauner & Haller, 2004. However, also authoring on a desktop PC is supported by CATOMIR. It is based on a visual programming interface approach, where the author can drag-and-drop components from a pool and connect them to define the logical behavior. The authoring tool supports a three-step authoring approach, thus finding the right components for the application, tweaking the components respectively, and connecting components to define the logical behavior of the scenario (cf. Figure 3).



(a) (b) (c)

Figure 3: The three-step authoring approach of CATOMIR.

Components can be described with smaller components and the logical connection network between them. However, some components cannot be split into smaller ones, because of performance reasons or just because there are operations that are easier to be realized by using a native programming language (namely native component). During the design process of AMIRE ES, we had to guarantee that all those native components would be implemented on the new mobile platform to achieve the same behavior on the authoring platform as well as on the target platform.

In AMIRE all authored MR scenarios (or applications) are stored in an XML format to ensure the portability. Our approach is similar to the application description language APRIL introduced by Ledermann & Schmalstieg, 2005. After the creation of a scenario, the corresponding XML file will be executed on the different target platforms accordingly. Beside the application description, also additional authoring related meta-information can be stored, such as layout configuration tags. This enables the author to easily switch between several authoring tools, though authoring on a powerful desktop PC was often the default case.

4 FaiMR – the evaluation scenario

As a demonstration for the test and the flexibility of the AMIRE ES environment, we decided to implement FaiMR, a Furniture Assembly Instructor program based on MR technology. The motivation for this program was the fact, that complicated manuals or instructions are a common problem when assembling furniture. Moreover, most of those manuals are just another unwanted annoyance. Antifakos, Michahelles, & Schiele, 2002 showed a very innovative idea that involved placing multiple sensors onto different parts of the assembly. As a result, the system as a whole could recognize user actions and thus monitor the whole assembly process. Our approach, likewise, uses Augmented Reality technology to solve assembly problems. By using ARToolKit, developed by Kato, Billinghurst, Blanding, & May, 1999, we attach reference markers to the various furniture parts, thus allowing a system to determine the point and order of assembly. Augmented, highlighted boards help users to find the right boards and to mount them in the right order. Thus, users get information where to place the different components and how to connect them.
FaiMR was first presented at SIGGRAPH 2003 (Zauner, Haller, Brandl, & Hartmann, 2003), followed by a presentation at ISMAR 2003 in Tokyo (Zauner, Haller, Brandl, & Hartmann, 2003).

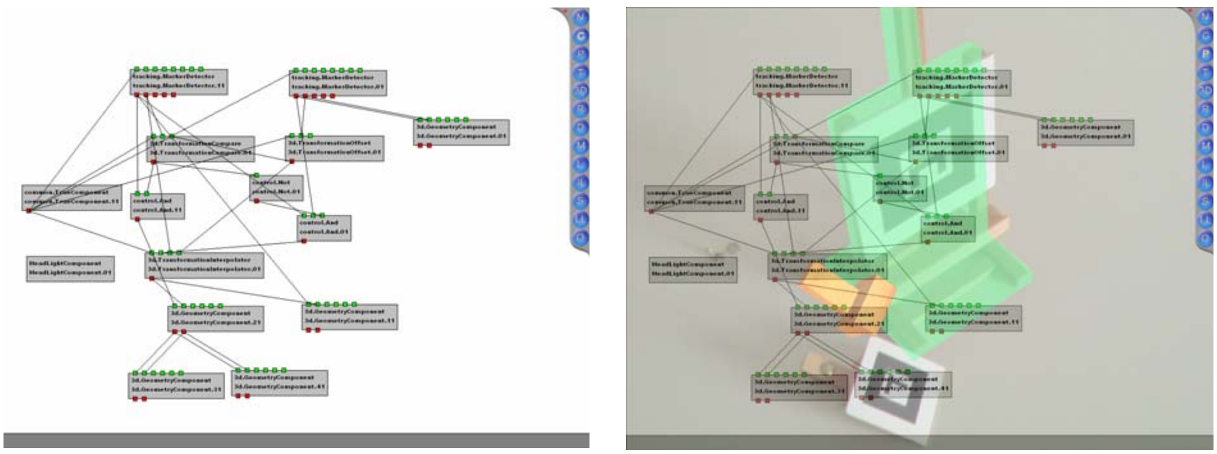
(a) (b) (c) (d)

Figure 4: Using FaiMR, users get a step-by-step instruction about how to assemble complex objects.

The first version was very promising but only available for PC (cf. Zauner, Haller, Brandl, & Hartmann, 2003). It was not authored with the AMIRE tools, but implemented in four months based on the AMIRE component framework, because at that time the authoring tools were not finished yet. Based on the first version of FaiMR, we authored the application again by using CATOMIR's visual programming approach. We wanted to know, how long it would take to implement the same application by just using our authoring tools. Even though we added new components for special features, the whole authored application was finished in about three weeks. This version was mainly designed for a Tablet-PC setup. Finally, the authoring of the PDA version of FaiMR (which was a light version of the 2nd version) was completed in about two hours including the modelling task, but without the port of the AMIRE framework to the PDA version. Beside the setup of the application, authors can easily change the application and tweak component's properties respectively. Since the Tablet-PC version differs from the PDA version, a one by one usage of the same XML-file was not possible. But since both applications have the same geometry files, including the same behavior and the same components, this would be possible as well; thus, authoring the application once with CATOMIR and execute it on both systems. Both applications the Tablet-PC version and the PDA version are very similar regarding the input and output metaphors that have been used in FaiMR. Consequently, the author could assume that after the authoring of the Tablet-PC.

5 Results and Discussion

The Tablet-PC version of FaiMR was more complex compared to/ versus/ vis-à-vis the PDA version which included xx components (cf. Figure 5). We used the Tablet-PC Compaq TC1100, equipped with 256 MB, 800 MHz, and a Geforce 4 and a PDA HP iPAQ 5550 with 128 MB memory.



(a) (b)

Figure 5: The component network can either be visualized without the background or with the augmented background, where the author can see the real world.

Regarding the performance, our PDA frame grabber delivers four to five images per second with a resolution of 320x240 pixels and a color depth of 16 bits compared to the Tablet-PC, where it delivered 13 to 27 images per second (resolution of 1024x768 pixels and color depth of 32 bits). We did not get a worse performance by using the AMIRE framework and including the overhead for the component approach – on the contrary, we were surprised about the good performance we got.

Detecting the ARToolKit markers, we achieved an average frame rate of about 10 to 15 fps for the Tablet-PC and 4 to 5 fps for the PDA, compared with no marker detection, where the Tablet-PC runs with 29 to 30 fps and the PDA with 6 to 7 fps.

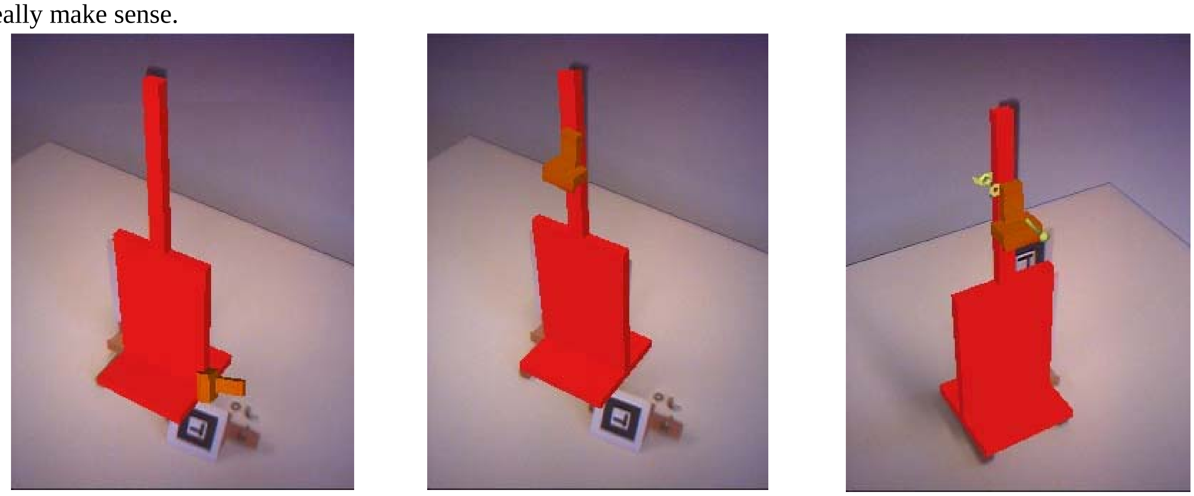
Thus the marker detection requires approximately 50% performance and 30% performance respectively. The optimized ARToolKit modification runs natively on an iPAQ at about 10-15 frames per second (cf. Wagner & Schmalstieg, 2003).

Regarding the 3d models, we used nine models with 1750 triangles and 34 textures for the Tablet-PC version and five models with 368 triangles and no textures for the PDA version. As depicted in Figure 7, the graphics quality for the PDA allows a shaded object with a final frame rate of about five frames per second and a model complexity of about 400 polygons, Gouraud-shaded and with one light source.

Consequently, the AMIRE framework works fine for a PDA without any hardware acceleration card – even though, with a reduced complexity of the polygons regarding to the Tablet-PC version. For now, a major drawback is the performance of the hardware. Several manufacturers of PDAs announced camera support and hardware support for 3d graphics. Therefore, the system will improve in performance and applicability, so that the PDA based assembly instructor can also be used in other application fields like, for instance, assembling large machines.

Figure 6: The PDA version of FaiMR shows a simplified step-by-step assembly instruction example.

Due to the small display and the low display resolution of the PDA, we recognized that the textured objects do not really make sense.



(a) (b) (c)

Figure 7: Figures (a) an (b) show the animation of the component that should be placed onto the easel.

Thus, the work we have invested in porting the image library DevIL for the PDA did not grow/ become/ was not disproportionate to the benefit we expected to achieve. On the contrary, the Gouraud shaded objects with well defined colors helped people to better recognize the augmented object instead of the seamless integration of the textured, augmented object.

Both applications have been authored with the same authoring tool. We used the light version CATOMIR, which allowed an easy and quick authoring for the PDA version – assumed, that the corresponding components already exist for the PDA platform. Once the application has been finished (either for the PC or the Tablet-PC), it is really simple to configure and/or re-configure the existing application to adapt it for the mobile platform and it would definitely absorb more time to do it by coding the application. Especially for a fine tuning, the authoring tool was really convincing and we did not want to miss it for the calibration step.

5.1 Conclusions

We extended our AMIRE approach and implemented additional features to support a mobile version for authoring MR applications that can be executed on PDAs respectively. The goal was to offer an environment that makes it possible to author an MR application once and to switch easily to another in/output system (e.g. to a PDA-version). We have chosen a PDA because of three reasons: firstly, because the PDA setup is a really challenging environment for the AMIRE framework in respect of the performance and resource limitations. Secondly, because the display is very small and some trade-offs in the design for the GUI have to be considered. In our case, we didn't support the whole palette of GUI as in the Tablet-PC version. Moreover, some features that are designed for the Tablet-PC version do not make the sense for the PDA version (e.g. animations because of performance and display limitations, textured objects etc.). Thirdly, because the interaction metaphors for the Tablet-PC are comparable to those for a PDA.
Summarizing, we can say that our AMIRE ES framework allows a fast and easy implementation for MR applications for both, the Tablet-PC and the PDA platforms. We realized a complex application to demonstrate the whole authoring process and where surprised about the fast results, people with no experience can achieve. Finally, we want to develop more features for the PDA version (e.g. integration of speech recognition, sound in- and output, etc.) and start with usability tests to improve the current authoring system.

5.2 Acknowledgement

We would like to thank Dieter Schmalstieg and his group for their support regarding the KLIMT software. In particular, we thank Daniel Wagner for his useful input regarding the integration into AMIRE ES framework. The AMIRE project was partially funded by the European Commission as part of the IST program within the 5th framework (project no. IST-2000-34024).

References

Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riß, S., Sandor, C., & Wagner, M. (2001), Design of a Component-Based Augmented Reality Framework, *Proceedings of The Second IEEE and ACM International Symposium on Augmented Reality (ISAR 2001).*

Piekarski, W. & Thomas, B. H. (2002). ARQuake: The Outdoor Augmented Reality Gaming System. *ACM Communications*, Vol. 45, No. 1, pp 36-38, 2002.

Abawi, D., Dörner, R., Haller, M., & Zauner, J. (2004). Efficient Mixed Reality Application Development, *1st European Conference on Visual Media Production (CVMP),* pp. 289-294, IEE, London, March 15, 2004.

Broll, W., Lindt, I., Ohlenburg, J., Wittkämper, M., Yuan, C., Novotny, T., Mottram C., Fatah A., & Strothmann, A. (2004). "ARTHUR: A Collaborative Augmented Environment for Architectural Design and Urban Planning", *in Proc. of the fourth International Symposium on Human and Computers (HC 2004)*, pp. 102-109.

Zauner, J., Haller, M., Brandl, A., & Hartmann, W. (2003). Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures, in *Proc. of the Second International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, Tokyo, Japan, October, 2003.

Zauner, J. & Haller, M., (2004). Authoring of Mixed Reality Applications including Multi-Marker Calibration for Mobile Devices, *in Proc. of 10th Eurographics Symposium on Virtual Environments (EGVE 2004)*, Grenoble, France, June 8-9, 2004, pp. 87-90.

Zauner, J., Haller, M., Brandl, A., & Hartmann, W. (2003). Authoring of a Mixed Reality Furniture Assembly Instructor, in *ACM SIGGRAPH 2003 Conference Abstracts and Applications*, San Diego, CA, 2003.

Kato, H., Billinghurst, M., Blanding, B., & May, R., (1999). ARToolKit, Technical Report. Hiroshima City University. December 1999.

Wagner, D., Schmalstieg, D. (2003). First Steps Towards Handheld Augmented Reality, In *ISWC*, 2003.

Haller, M., Zauner, J., Hartmann, W., & Luckeneder, L. (2003). A generic framework for a training application based on mixed reality. *Technical report*, Upper Austria University of Applied Sciences (MTD), 2003.

Ledermann, F., & Schmalstieg, D., (2005), APRIL - A High Level Framework for Creating Augmented Reality Presentations, *In proceedings of IEEE Virtual Reality 2005*, Bonn, Germany, March 12-16, 2005

Antifakos, S., Michahelles, F., and Schiele, B. (2002). Proactive Instructions for Furniture Assembly. *In Proceedings of the The Fourth International Conference on Ubiquitous Computing* (UbiComp 2002).