

# AMIRE – Authoring Mixed Reality

Paul Grimm  
Fraunhofer AGC  
grimm@agc.fhg.de

Michael Haller  
FH-Hagenberg  
haller@fh-hagenberg.at

Volker Paelke  
Uni Paderborn  
vox@c-lab.de

Silvan Reinhold  
Fraunhofer AGC  
sreinhold@agc.fhg.de

Christian Reimann  
C-Lab  
reimann@c-lab.de

Jürgen Zauner  
FH-Hagenberg  
jzauner@fh-hagenberg.at

**Keywords:** Mixed Reality, Authoring, Production Process

## I. OVERVIEW

This paper describes the requirements of a training demonstrator based on Mixed Reality in the context of the AMIRE project. The overall objective and idea of the AMIRE project is to define and implement a software system that allows content experts to easily design and implement mixed reality applications without detailed knowledge about the underlying base technologies of MR. Therefore, it describes a generic framework that allows an easy communication between the objects (MR components) used in the AMIRE applications.

The AMIRE framework uses a component oriented technology and consists of a the minimal set of components required for a demonstrator, a reusable GEM collection and a visual authoring tool for building MR applications. For the realization of the AMIRE framework we also require an object tracking system that is cheap and easy to use. These are features of the ARToolKit library [1]

Most of the current MR applications focus on the development of AR base technologies, like a lot of projects in the area of research and development of AR and MR applications. Many European projects mainly focus on the development of MR applications for a special domain (e.g. technical maintenance), such as ARVIKA, STAR, etc. [4]. But none of these projects focused on a structured authoring approach with reusable components [3] and GEMS. Due to the complexity of MR/AR applications only MR experts are able to develop MR/AR applications and prototyping of MR based applications becomes a very difficult task, because most research institutes have to develop these applications from scratch.

## II. THE AMIRE PRODUCTION PROCESS

The main production process comprises several tasks, ranging from the recognition of the actual problem to the final solution in the form of an AR application:

**Conception of the Storybook.** This task includes a thorough analysis of requirements and is implemented mainly by the domain expert. It leads to the

**Identification of necessary Components.** Components that are required for the implementation are qualified by the author and must then be supplied in the process of

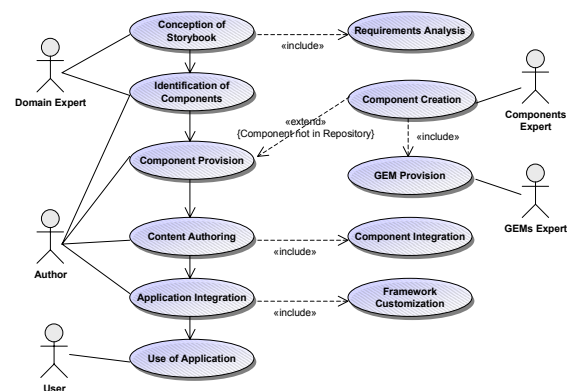
**Content Provision.** The author searches the repository for available Components; those that do not exist must be created by the Component expert, who in turn must work with the GEMs expert to assemble Components. When all Components have been provided, the author begins with

**Content Authoring.** This includes the integration of all Components, i.e. adaptation and combination with other Components. Finally, the author must perform the step of

**Application Integration.** The application must be made available to the user within the runtime framework.

**Use of the Application.** The end user can now operate the application; development cycles are of course possible and intended, taking into account that an application goes through an evolutionary process, needs maintenance, bug fixes, revision of concepts etc.

The relationships between these tasks and the actors who partake in this process are shown in Figure 1.



**Figure 1** The AMIRE production process

## III. THE AMIRE FRAMEWORK

The framework is the glue between GEMs and components. It holds everything together and supports component developers as well as gem developers to design new components or GEMs.

Component developers must use the generic uniform component interface, which is part of the framework. It supports the generic configuration of components and the communication between components. Further it allows the application programmer or author to make component instances persistent.

The framework also provides functionality for gem developers. The provided interfaces are not as generic as the component interfaces. But in combination with conventions suggested by the framework GEMs will be integrated fast and easily into the framework. Therefore plug-in interfaces will be integrated into the framework. This will make it as easy as possible to change from one gem to another gem, which provides the same functionality at another license policy or performance rate.

Mixed Reality components in AMIRE are the essential elements in the authoring process. Components represent solutions for particular domain specific problems and typically combine and extend AMIRE GEMs towards advanced high-level features of a MR application. Components are defined as domain specific elements. From a high level perspective, they are structured in geometry models and behaviour. Behaviour could comprise animations or simulations of behaviour.

Some of the desirable features for AMIRE components would be the following: Structured, Reusable in different versions, Reusable in different applications, Extensible, Flexible. Besides their domain-oriented design, some components may be useful for several domains due to its generic characteristics.



**Figure 2: Virtual Glovebox tutorial as AMIRE demonstrator**

#### IV. MR GEMS

Many small tricks, ideas and solutions to problems that reoccur frequently in the development of MR applications have already been developed in other projects. Because such “small” ideas neither warrant publication in a research journal nor the creation of specific libraries they are usually unavailable to developers, so that solutions have to be reinvented again and again.

GEMs provide the low-level mechanism for the reuse and integration of such existing techniques and algorithms into the AMIRE framework. The concept of GEMs was inspired

by the “Graphics GEMs” series of books that was started in 1990 by Andrew Glassner. The vision and purpose of the series is to provide directly applicable tips, techniques, and algorithms for graphics programmers that were derived from practical experience.

In a similar fashion the AMIRE project will establish a MR GEM collection of techniques, algorithms and code snippets with efficient solutions to common programming problems in mixed reality applications, e.g. pattern based object recognition, path animation, 3D object loaders or AR specific image filtering techniques. The MR GEM collection in AMIRE will not only collect and catalogue existing solutions but also extend them with meta-information so that developers can easily identify suitable GEMs for a particular programming problem. The MR GEM collection will be used within the AMIRE project for the implementation of higher-level entities like components and the AMIRE framework itself, and will later also become accessible for use in other mixed reality projects.

#### V. SUMMARY

The AMIRE GEM collection aims to provide a useful ARToolkit repository for the MR developer community. Therefore, contributions of tricks and solutions from developers outside the AMIRE context are actively encouraged to cover the diverse needs of MR application developers.

#### VI. ACKNOWLEDGEMENTS

The work described in this paper has been undertaken in the context of the AMIRE project that consists of the following partners: c-Lab, FAW, Fachhochschule Hagenberg, Fraunhofer AGC, Helsinki University, Labein, OMV, Talentcode. 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 programme (project 34024).

#### VII. REFERENCES

- [1] Kato H., Billingham M., Blanding B., May R., *ARToolKit*, Technical Report, Hiroshima City University, December, 1999.
- [2] Feiner S., Höllerer T., Gagas E., Hallaway D., Terauchi T., Güven S., and MacIntyre B., *MARS - Mobile Augmented Reality Systems*, Columbia University Computer Graphics and User Interfaces Lab, 2001.
- [3] <http://www.mr-system.co.jp>  
Friedrich W., *ARVIKA Augmented Reality for Development, Production, and Service*, Tagungsband des Informationsforum Virtuelle Produktentstehung (IVIP), May, 2000.