EUROGRAPHICS Workshop on Multimedia (2004)
N. Correia, J. Jorge, T. Chambel, Z. Pan (Editors)

A Component-based Authoring Environment for Creating Multimedia-Rich Mixed Reality

Daniel F. Abawi[1], Ralf Dörner[2], Paul Grimm[3]

[1]Johann Wolfgang Goethe-University, Frankfurt, Germany
[2]Harz University of Applied Sciences, Wernigerode, Germany
[3]Erfurt University of Applied Sciences, Erfurt, Germany

Abstract

Applications that seek to combine multimedia with Mixed Reality (MR) technologies in order to create multimedia-rich MR environments pose a challenge to authors who need to provide content for such applications. Founded on a component-based authoring paradigm, production processes as well as tools that serve as a supportive authoring environment for these authors are presented in this paper. For this, not only requirements that stem from multimedia authoring or MR authoring alone have been identified but also authoring tasks that are only present in the creation of multimedia-rich MR content. Concepts for supporting these tasks within a component-based authoring framework (e.g. the specification of phantom objects) are presented. The resulting authoring tools are discussed - one of their main advantages is that they provide a direct preview of the content for the author. This allows multimedia authors who are not familiar with MR methodologies to quickly gain experience with multimedia-rich MR content creation.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities

1. Introduction

Based on computer vision and computer graphics, Mixed Reality (MR) methodologies allow to combine computer-generated images with images from the real world in an elaborate way. As a result, a blend of reality with virtual worlds can be achieved [MTUK94]. Hardware required for MR is becoming commonly available – simple MR installations need a multimedia PC with a webcam. With hardware readily available, MR methodologies have matured enough that first prototype applications of MR have been developed in several areas, for example e-learning, CSCW, storytelling or entertainment [SHM*02]. In order to create a richer MR experience and to make MR more convincing, multimedia elements are more and more integrated in MR applications [KNEO01, PCF*02b].

The combination of MR and multimedia drastically increases complexity and surpasses the complexity of multimedia systems or MR systems alone. As a consequence, such combined MR and multimedia systems do not only challenge system designers and developers who need to think of new models how to combine MR, multimedia and hypermedia [RC03] but also authors who are supposed to develop content for such applications. This is a serious obstacle for the advancement of multimedia-rich MR technologies since feasibility of content production is a key for a widespread usage of these technologies in applications.

Division of work and specialization are strategies that have been already employed successfully in multimedia production. Only few individuals possess all the skills and know-how needed in order to produce multimedia content - let alone multimedia-rich MR content. Thus, several authoring roles need to be distinguished and a production process needs to be carefully conceived. This production process should describe how the different author roles co-operate efficiently to create content. Moreover, supportive authoring tools are needed. These tools have to satisfy the same requirements as multimedia authoring tools, for example the provision of synchronized views especially when dealing with time-based multimedia [JRT00]. Here, one requirement is of particular interest: offering a direct preview of the content. For multimedia authoring tools this has been important since multimedia designers need to quickly explore and get feedback on their ideas in order to be creative [BKC01]. For authoring tools intended for a combination between multimedia and MR this is even more important: MR is a novel and only partly mature technology - few authors have experience which ideas will work and which won't. An authoring tool that provides an integrated preview and allows the author to switch directly to the user's point of view will accelerate learning processes and increase the gain of experience.

With the combination of multimedia and MR, novel authoring tasks have to be performed, e.g. the specification of spatial aspects of multimedia (like 3D sound or sensitive areas) or the association of single media objects to virtual or real parts of the world. An authoring tool for multimedia-rich content has to support these tasks beside the MR-related tasks like calibration. Few authoring tools and authoring concepts have been proposed for MR in literature [GPRR00, Fis02, PTB*01]. They are not particularly suited to integrate multimedia in MR and focus on the support of MR-related authoring tasks. Most of them do not offer direct preview capabilities and, in this context, lack concepts to integrate software components needed for multimedia (like plug-ins to play, encode or decode multimedia content).

In this paper, we present an authoring environment (i.e. authoring tools and production processes) that is suited for the creation of multimedia-rich MR content. This authoring environment uses an authoring paradigm that is based on the concepts of components and frameworks. For the realization, we use the AMIRE (Authoring Mixed Reality) [AMI04] authoring framework that is dedicated to MR authoring and introduce new components that support the tasks associated with the combination of MR and multimedia. The resulting authoring environment can in turn be advantageous to multimedia content creation in general since it points out new approaches to improve multimedia authoring tools [ADG*02]. MR offers the potential to create novel user interfaces [BKP01b] and has already been used to improve authoring tools for media, e.g. for video editing [MP94].

This paper is organized as follows. In section 2 we identify issues and tasks related with the creation of multimedia-rich MR content. Section 3 presents our component-based authoring paradigm and our authoring processes. In section 4, we present how the AMIRE authoring framework implements according components and functionality for supporting the authoring tasks identified in section 2. Finally, section 5 gives a conclusion.

2. Multimedia and MR

Generally, a MR application is based on a run time environment which handles all MR-specific tasks like the integration of computer vision algorithms or the (marker-based or marker-less) tracking system. Consequently, it is responsible for the mapping between virtuality and reality.

Beside the overall behavior of a MR application the integration of multimedia content is essential. Multimedia-rich applications are characterized by various interactions of several single media objects. First of all each media object has to fulfil the requirements concerning expressiveness, quality and complexity [SN04]. Moreover, it is crucial for multimedia-rich applications to specify requirements for synchronisation between all media objects, e.g. to describe dependencies between video, sound and text. Furthermore, the spatial aspect for each media object has to be specified in the MR application. This specification includes the spatial relationship between several media objects in a three-dimensional application space as well as their relationship with corresponding objects in the reality.

Several types of media objects can be used to build MR applications. The following overview identifies issues related with the usage of media objects in MR applications:

(Hyper)-Text: E.g. text can be used to augment the reality with additional information. In addition, hypertext can be used for navigation and interaction purposes, where real and virtual objects can act as links. In [KLK*02] text is used to support a user for indoor pathfinding.

Images and Textures: Using images MR applications can exchange or augment parts of real worlds, e.g. they can be used to exchange paintings or to add advertisements to an application. Textures can be used to change the appearance of real objects within a MR-application, e.g. to change the appearance in architectural arrangements [PT02b]. The congruency of the texture used and the real object is crucial for the user acceptance.

Video: A video can be placed on any plane. E.g. video can be used to show an arbitrary channel on a television set or it can be used to build a MR video-conferencing tool [BKP01a].

3D Video: In multimedia-rich MR applications the user controls his own field of view usually without constraints. Consequently, a 3D video can be seen from any angle. [PCF*02a] has shown how 3D video can be build and integrated in a MR application. They use a system with 15 cameras to record the 3D video. The presented system was used exemplary to transmit scientific talks and discussions (see Figure 1).

Interaction Elements: [PT03] shows an example how interaction elements can be integrated in a MR application. They are used for color specification.

(3D) Sound: Within a MR application a sound can be associated with a specific 3D position. Therefore, it is necessary

Figure 1: *3D Video in a MR Application [PCF*02a].*

to simulate effects like attenuation or echo. In MR applications it is possible to simulate echo with phantom objects (i.e. virtual objects that represent real objects in the virtual model of the scene), which affect the propagation of sound waves like real objects would do. [DHS02] describes authoring tasks related with inserting 3D sounds in a MR scenario.

3D Geometries: 3D geometries are used to place three-dimensional virtual objects into reality. Therefore, mechanisms are needed to allow an occlusion of a virtual object by a real object. This task can be done using occluder objects, which are inserted in the virtual scene of a MR application. Each occluder object corresponds to a real object and enables occlusion calculation between them. To accomplish the perceptual requirements of depth cueing it is recommended to integrate also shadow calculation into MR scenes [HDHZ03] (see Figure 2). Therefore, it is necessary to insert phantom objects, on which the shadow can be projected.

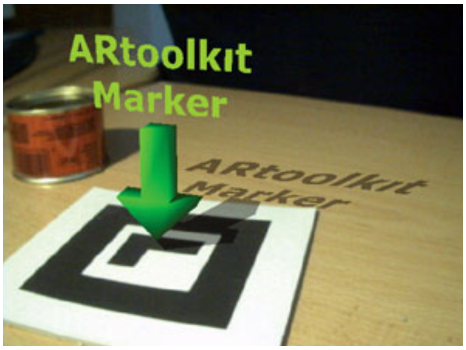

Figure 2: *Shadows in MR applications [HDHZ03].*

3D Animations: It is also possible to animate 3D geometries. [PT02a] and [CGF*04] have build MR games using 3D animations. To avoid inconsistencies it is necessary to use algorithms for collision detection between virtual and real parts or collision avoiding. Therefore, the basic algorithms have to be integrated into the MR run time environment. In addition, it is the task of each author to define the reactions on the events emitted by these basic collision detection algorithms.

To allow a positioning and orienting of multimedia objects within a MR application and to accomplish the perceptual requirements of depth cueing (like shadow and occlusion calculation) for all multimedia objects it is necessary to handle them like 3D objects. Consequently, the author has to define phantom objects for multimedia objects where MR specific aspects like shadow calculation or occlusion should be considered by the system.

The prerequisite for using any media objects in a MR application is to produce it. If the object needed is not available in a local repository or a commercial library (or may be too expensive) it is necessary to build it. We propose to use common media-specific tools (like Adobe Photoshop to create images or Alias Maya to build 3D geometries) to produce single media objects, because these tool are widespread and their methodologies are well known.

To collect and integrate the produced single media objects to be used in the MR application we propose to use a MR specific authoring environment. With media-specific tools it is difficult to solve all identified problems like occlusion, congruency, specification of geometric and semantic dependencies (e.g. synchronisation of different media objects with events caused in the reality), collision detection and integration of phantom objects for sound and shadow calculation. Therefore, we introduce in the next section a MR specific authoring process with corresponding phases and tools.

3. Authoring Process, Phases and Tools

Due to the complexity of MR methodologies, a broad spectrum of software developers and content authors need to contribute to MR for various application areas. As a consequence, potential MR authors are not a homogeneous group, but they have significantly different requirements and skills. To address this 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 (Figure 3). Since no single person usually possesses all the expertise required, different author roles need to be distinguished.

On the author role level, therefore, it would be beneficial if the author roles were arranged in the form of an authoring pyramid (Figure 3). Authors at the top of the pyramid (like domain experts) would be able to benefit from the products of authors who deal with technically more difficult tasks, but are able to perform their tasks without direct involvement of authors from a lower level of abstraction.

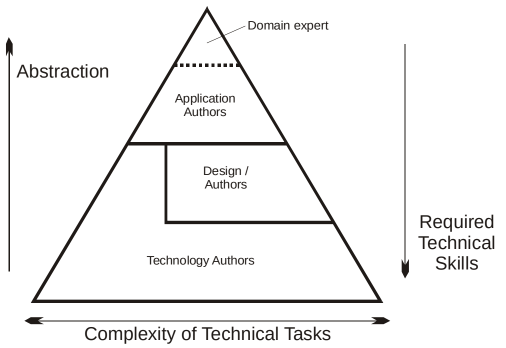

Figure 3: *The authoring pyramid, different degree of technical abstraction.*

Our approach is to build an architecture that makes available authoring tools that are individually tailored to suit individual authors (e.g. by providing varied degrees of technical abstraction). For this, a subset of existing authoring functionality is individually chosen by an author whose role is the configuration of authoring tools for higher-level authors. 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 (cf. section 3.3).

The main idea in our approach is to let the author assemble a MR application by reusing and adapting predefined building blocks (i.e. domain-unspecific building blocks that can be filled with domain-specific content, cf. section 4).

On the production process level, there is no established way how such a complex system is to be build and how the different authors should co-operate. Even the roles are not clearly defined. Established forms of creating content, e.g. multimedia content like videos or 3D animations, cannot be transferred easily to MR content production. In addition there are MR-specific tasks to accomplish (cf. section 2). As a consequence 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 the author omits no phases needed.

We combine the successfully used approaches to employ software libraries [HCF97] on the technology level and to use MR methodologies themselves to support authoring tasks [Fis02, HFT*99]. Using MR during the development simplifies complex tasks mentioned in section 2, like the integration of multimedia elements (cf. section 4). Using a software framework greatly reduces the complexity of MR application development and has several additional advantages (like increasing efficiency, extensibility and adaptability) [HCF97].

3.1. Authoring Phases

When following a framework-oriented approach, building a MR application using existing components involves four 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 virtual objects represented in the MR components

In the qualification phase (cf. Figure 4), 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 reuse engineers in the context of thorough requirements engineering.

During the adaptation phase (cf. Figure 4) the author has to customize the MR components in order to fulfil the particular requirements posed by other building blocks and by the application itself. As the individual buildings blocks are usually created with different application contexts in mind, this 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 4) the author defines the relationships and interaction mechanisms between a set of MR components. This task is technically supported by embedding all MR components in a common MR run time framework. All MR components have according interfaces that facilitate communication with other MR components.

Our development process has to reflect and adequately address the specific requirements of MR identified in section 2. 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 with a real chair).

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

The authoring tools have an important role in the authoring process, since the author requires a tool that allows him

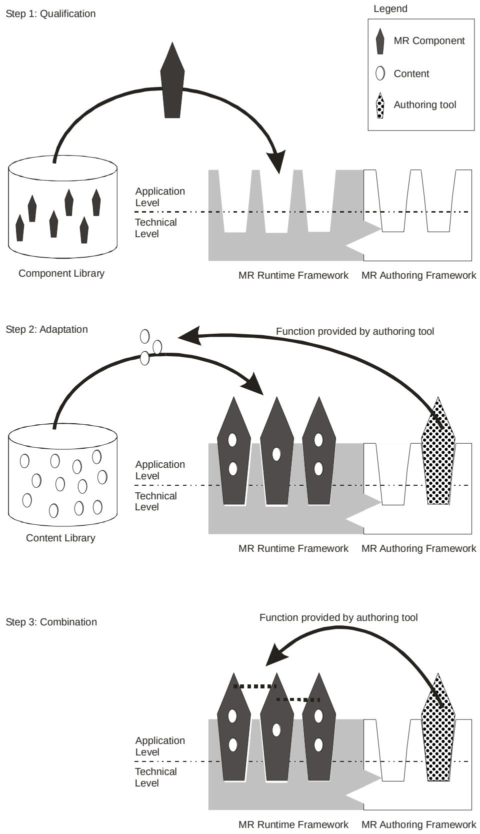

Figure 4: *Three phases during the development process.*

to complete his individual authoring task. In our architectural concept, an authoring tool is obtained by coupling a MR authoring framework with the MR run time framework in which the MR components are embedded. The authoring framework can be customized with specific authoring tool components. Such authoring tool components exist for different authoring phases. Like the MR components, they can be customized in order to meet the requirements of different author groups. Alternatively, different authoring tool components are plugged 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 in our architecture, the authoring tool components can interact with MR components and thus realize functionality like the qualification of MR components.

3.2. Using a Mixed Reality View

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. As a consequence, the AMIRE authoring environment shows the end-user view during the authoring time (cf. section 4), that can be extended with authoring specific information (cf. Figure 13). This ensures that the pre-view of the author and the view of end-user can be made identical without any additional efforts.

We found that authoring MR applications is easier and more efficient, if the results can be directly viewed without any transformation steps. Our approach reflects this issue by using (technically) the same software framework to build and run the MR application (cf. Figure 4). The authoring framework bases on the technical (run time) framework and extends this by specific authoring tools that encapsulate the functions provided by the technical underlying framework for the author [ZHHL03]. The framework itself is based on an approach that employs software components.

This benefit of using MR during MR authoring is amplified by the possibility to allow offline authoring. Still the advantages mentioned before are available, but instead of using real-time video input-devices, pre-recorded video streams enable the author to preview his MR application under different conditions (e.g. lighting conditions) - even without being physically present at a certain location in reality.

3.3. Extensibility of the Toolset

Our development process does not limit the author to reuse predefined blocks. Depending on the author's skills (cf. Figure 3) he is able to create new reusable blocks or content that can be filled into these blocks.

In order to provide domain-specific components, our approach is to make our architecture extensible and to ensure that users can enter the development process on different levels. Figure 5 illustrates the association of the elements of our architecture to different authors or user groups respectively.

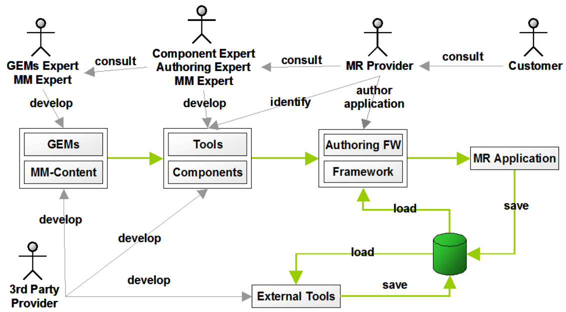

Figure 5: *MR application development workflow.*

If the author could not find suitable existing components, he is able to develop his own components or even dedicated authoring tools. Less technically orientated is the possibility to integrate content (like multimedia elements) in the predefined building blocks or to create respective model content that could be filled into predefined blocks. These steps can be allocated to different authors, likewise the different phases can be allocated to different authors.

The development process should consider that MR is depending on the reference points in real space, therefore the MR applications should easily be maintainable and extendable. 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.

Therefore, we reference the content elements (like multimedia files) rather than integrate them statically. The constructed MR scene is described in a specific XML-language, which includes a description of the content elements needed. These content elements are loaded during run time (cf. section 4). This allows us to integrate author-specific third party applications into the development process ([ADG*02]).

4. MR Authoring of Multimedia Elements in AMIRE

The following use case shows how an author is supported by the AMIRE authoring environment while developing a MR application with multimedia elements.

In our use case, we sketch the necessary steps to build a MR application that augments a real showroom with a graphical multimedia element (i.e. we place a rich-media model of a Porsche of the year 1955 in an empty showroom). Beside the explanation of the usage of the AMIRE authoring tools that support the four phases mentioned in section 3, we present dedicated tools that handle the specific problems posed by integration of multimedia elements (cf. section 2). For instance, the showroom in our use case contains a pillar that have to be taken into account in respect of the aspects mentioned in section 2.

Figure 6 depicts the authoring environment of AMIRE. The workspace contains, as mentioned before, the end-users view of the MR application (from now on called video image). Additionally, AMIRE includes some standard dialogs that we will describe step by step. The environment of AMIRE with its windows and position is customizable by the author. According to his preferences AMIRE allows him to add new authoring tools and to position them in the workspace.

Figure 7 shows the window that lists existent MR components, which are present in the directory and hierarchically sorted. The author may select reusable MR components that are needed to assemble the MR application. In order to display multimedia content, 3D geometries, or text in a scene, the author might include a placeholder object such as an image loader, a geometry loader (as depicted in Figure 7) or a

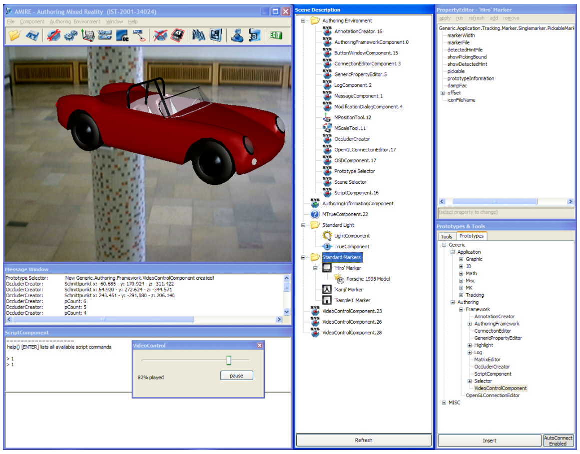

Figure 6: *The AMIRE authoring environment.*

text component. Such components allow a quick exchange of their visual representation without removing the components themselves from the scene.

Figure 7: *Selection of predefined MR components in the qualification phase.*

As a MR component is usually not a special purpose object with respect to the actual context it will be used in, some adaptation will always be necessary. Figure 8 shows the property editor of AMIRE. In our use case the author has to specify the filename of the 3D model he wants to fill in the placeholder object. To support adaptation without revealing any implementation details, MR components have properties, which can be accessed through property editors. Figure 8 shows the properties of the geometry loader component. Since the AMIRE authoring tools are implemented as components either, the author carries out their adaptation in the same manner. Beside the adaptation of properties with

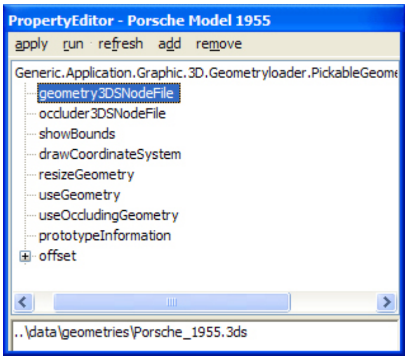

Figure 8: *Customizing MR components in the adaptation phase.*

the generic property editor, components can offer specific user interfaces with dedicated metaphors. We will see an illustration of such a tool later (see Figure 12).

In the combination phase, components can be connected to interact with each other. In AMIRE each MR component has so-called in-slots and out-slots, which are used for receiving and sending information, respectively. Connecting two components is simply a matter of connecting the out-slot of one component with an appropriate in-slot of another one. Figure 9 shows the connection editor of AMIRE; the geometry loader component (that loads the 3D model of the car) is getting connected with a component that supports playing of sound files.

To arrange position and orientation of virtual objects with the real video, tracking systems are essential for MR. AMIRE uses the optical tracking system ARToolKit [KB99]. In our approach, the markers are also represented as components. Thus, the geometry loader component that visualizes the 3D car model in our use case has to be connected with a specific marker component (cf. the marker on the pillar in Figure 15).

As mentioned in section 2, one crucial factor in MR is the alignment of virtual objects (i.e. multimedia elements) with reference points in real space. Figure 10 shows the tool that allows such operations. The author may change the position and orientation of virtual objects and perceives the end-users' view in the video image (cf. Figure 6). AMIRE offers different types of tools that allow a manipulation of the virtual objects' pose. Another authoring tool uses MR as interaction metaphor; the author indicates a position and orientation by moving a marker in reality [ADG*02]. In order

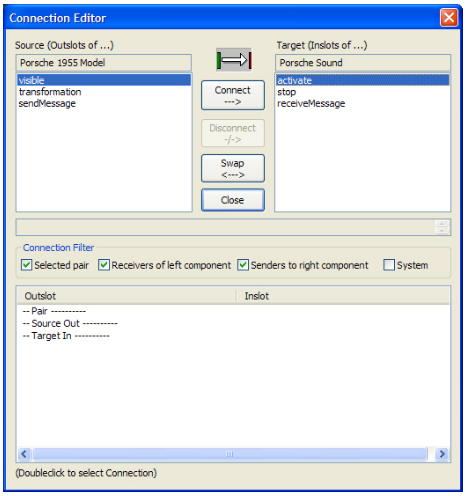

Figure 9: *Connecting MR components in the combination phase.*

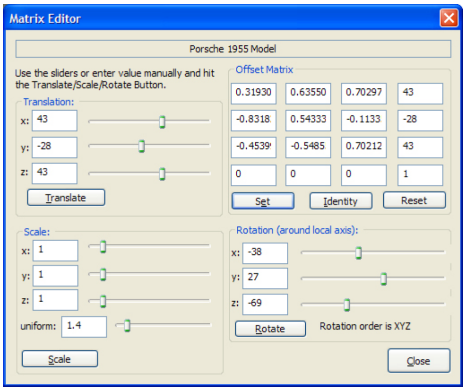

Figure 10: *Position multimedia elements in the calibration phase.*

to increase tracking performance, a virtual object can be connected to more than one marker (to a so-called multi-marker component). The virtual object is shown in the correct pose if at least one marker is detected correctly. AMIRE supports the author with dedicated tools that hide these technical aspects from the author.

One of the most important windows provides the author with an overview of the current elements in the scene (cf. Figure 11) sorted by different categories or in a hierarchy the author has specified. Thereby the author receives an overview of the scene and the (geometrical) relations between components. The author has the possibility to rename components and to regroup them via a drag-and-drop mechanism.

AMIRE supports specific aspects, like the creation of

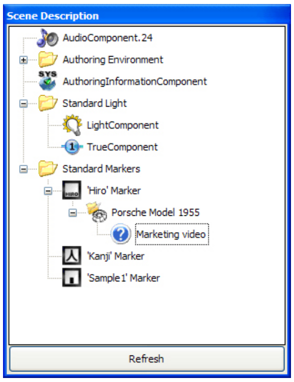

Figure 11: *A hierarchical view of the elements in the scene.*

phantom objects (Figure 12 right depicts a tool to create phantom objects) with dedicated tools. The author has the possibility to create phantom objects by specifying points (from different perspectives) in the video image; AMIRE creates phantom objects that occlude real objects in order to augment the scene correctly, cf. Figure 6 and Figure 15. Figure 12 (left) shows a tool that allows the author to create annotations, a widely used possibility of MR. The author specifies a text, color settings etc. Both tools in Figure 12 are components that have specific user interfaces that allow their manipulation, in addition to the possibility to use the property editor (cf. Figure 8).

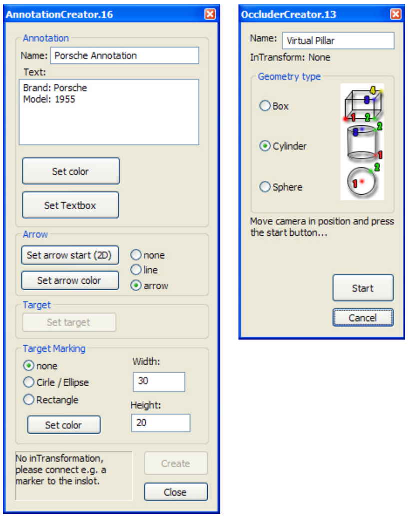

Figure 12: *Dedicated authoring tools with specific user interfaces.*

Some tools in AMIRE offer a view that should sensitize the author for specific MR problems. Figure 13 shows the view used to support the author detecting edges in order to specify the bounds of phantom objects. Other views are appointed to draw the author's attention to missing or wrong shadow settings of virtual objects (i.e. because of missing phantom objects; cf. section 2).

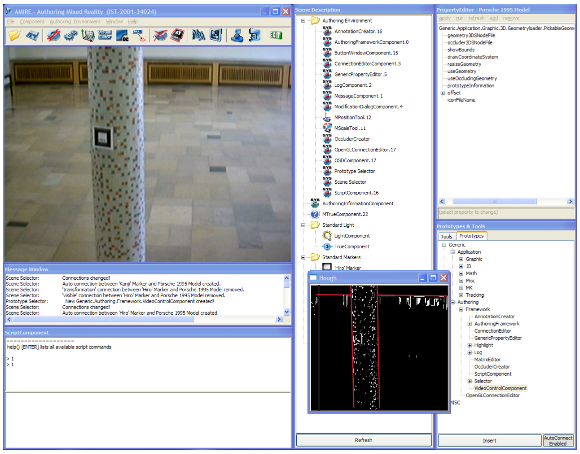

Figure 13: *Additional views to sensitize the author for specific MR problems.*

To support the author specifying the relationships between MR components we have introduced our connection editor (cf. Figure 9). Depending on a specific event, one component sends information to another one it is connected with. For events that depend on the position or orientation of the end-user (or the author) AMIRE supports the author during the specification of redemptional event. Figure 14 shows the visualization of an area sensitive component that triggers an event if the user enters the area. Beside the specification or adaptation by manipulating numerical values (e.g. the property editor, cf. Figure 8) AMIRE enables visualizations of relations between components in the MR scene.

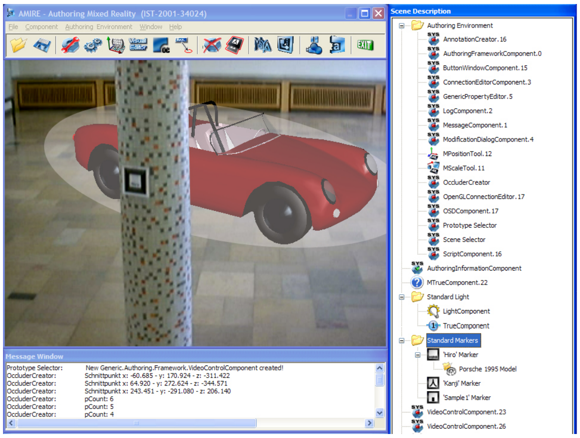

Figure 14: *Visualization of redemptional events.*

In our use case, we have specified that a sound should be played (e.g. the specific engine noise) if the users enters the area highlighted in Figure 14. Therefore, a component triggers the end-users' position (by tracking ARToolKit markers), parameterized with the according area.

In the AMIRE authoring environment, the author has at every point in time the end-users' view (enhanced with authoring specific elements) of the MR application. AMIRE support him with dedicated tools and a well-structured process. Once the steps above are carried out by the author, a simple MR application of our use case (cf. Figure 15) augments the showroom with a 3D model of the Porsche (with correct occlusion of the pillar). The author might change the multimedia element later on. All steps are carried out without programming skills. The description of how typical MR-

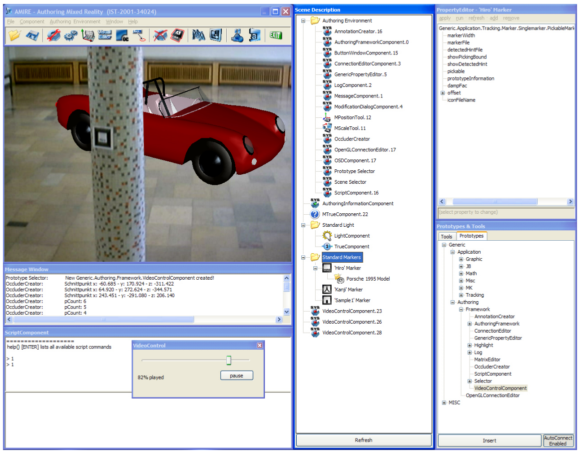

Figure 15: *A phantom object occludes the pillar.*

related authoring tasks can be supported by our toolkit shows the potential and the level of support authors can expect from our approach to MR authoring.

5. Conclusion

Authoring environments for creating multimedia-rich MR content can not be build by simply combining authoring tools for multimedia with authoring tools for MR. We identified requirements that stem not only from multimedia authoring or MR authoring alone but also authoring tasks that are only present in the creation of multimedia-rich MR content. For supporting authors a sophisticated authoring environment needs to be provided. We showed that a component-based authoring paradigm is suited to do this. Concepts for supporting specific authoring tasks (e.g. the specification of phantom objects) could be designed and implemented under this paradigm. One of the major advantages of the resulting authoring tool is that it provides a direct preview of the content for the author. This allows multimedia authors who are not familiar with MR methodologies to quickly gain experience with multimedia-rich MR content creation.

6. Acknowledgements

We would like to acknowledge the work and the valuable contributions of all partners in the AMIRE project. This work was partly funded by the European Commission under the IST-programme (IST-2001-34024). Many thanks to Frank Bergmann, Joachim Bienwald, Sönke Dirksen, André Miede, Michael Krauss, Björn Schmidt and particularly Jürgen Zauner for their contributions.

References

[ADG*02] Abawi D. F., Dörner R., Geiger C., Grimm P., Haller M., Hartmann W., Reinhold S., Zauner J.: Mixed reality - a source for new authoring methodologies? In *ACM SIGGRAPH and Eurographics Campfire: Workshop Proceedings Production Process of 3D Computer Graphics Applications - Structures, Roles and Tools* (June 2002), Shaker, pp. 9–16.

[AMI04] The AMIRE (authoring mixed reality, IST-2001-34024) project homepage, http://www.amire.net, 2004.

[BKC01] Bailey B. P., Konstan J. A., Carlis J. V.: DEMAIS: designing multimedia applications with interactive storyboards. In *Proceedings of the ninth ACM international conference on Multimedia* (2001), ACM Press, pp. 241–250.

[BKP01a] Billinghurst M., Kato H., Pair J.: Mobile mixed reality conferencing. In *2001 International Symposium on Mixed Reality (ISMR)* (2001).

[BKP01b] Billinghurst M., Kato H., Poupyrev I.: The MagicBook - moving seamlessly between reality and virtuality. *Computer Graphics and Applications 21*, 3 (2001), 6–8.

[CGF*04] Cheok A. D., Goh K. H., Farbiz F., Liu W., Li Y., Fong S. W., Yang X., Teo S. L.: Human Pacman: a wide area socio-physical interactive entertainment system in mixed reality. In *Conference on Human Factors in Computing Systems* (New York, NY, USA, 2004), ACM Press, pp. 779–780.

[DHS02] Dobler D., Haller M., Stampfl P.: ASR - augmented sound reality. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications* (2002), p. 148.

[Fis02] Fisher S. S.: An authoring toolkit for mixed reality experiences. International Workshop on Entertainment Computing, IWEC, 2002.

[GPRR00] GEIGER C., PAELKE V., REIMANN C., ROSENBACH W.: A framework for the structured design of VR/AR content. In *Proceedings of the ACM symposium on Virtual Reality Software and Technology* (2000), ACM Press, pp. 75–82.

[HCF97] HAINES G., CARNEY D., FOREMAN J.: Component based software development / COTS integration, 1997. Software Technology Review (draft), Revision 97a.

[HDHZ03] HALLER M., DRAB S., HARTMANN W., ZAUNER J.: A real-time shadow approach for an augmented reality application using shadow volumes. In *VRST 2003, ACM Symposium on Virtual Reality Software and Technology* (Oct. 2003), ACM Press, pp. 56–65.

[HFT*99] HOLLERER T., FEINER S., TERAUCHI T., RASHID G., HALLAWAY D.: Exploring MARS: Developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics 23* (1999), 779–785.

[JRT00] JOURDAN M., ROISIN C., TARDIF L.: Scalable toolkit for designing multimedia authoring environments. *Multimedia Tools and Applications 12*, 2–3 (2000), 257–279.

[KB99] KATO H., BILLINGHURST M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings of IWAR 99* (1999), pp. 85–94.

[KLK*02] KALKUSCH M., LIDY T., KNAPP M., REITMAYR G., KAUFMANN H., SCHMALSTIEG D.: Structured visual markers for indoor pathfinding. In *Proceedings of the IEEE First International Workshop on ARToolKit* (2002). Available as technical report TR-188-2-2002-13, Vienna University of Technology.

[KNEO01] KIYOKAWA K., NIIMI M., EBINA T., OHNO H.: MR Square: A mixed-reality meeting room. In *Proceedings of ISAR'01* (2001), IEEE Press, p. 169.

[MP94] MACKAY W., PAGANI D.: Video mosaic: laying out time in a physical space. In *Proceedings of the second ACM international conference on Multimedia* (1994), ACM Press, pp. 165–172.

[MTUK94] MILGRAM P., TAKEMURA H., UTSUMI A., KISHINO F.: Augmented reality: A class of displays on the reality-virtuality continuum. ol. 2351, Telemanipulator and Telepresence Technologies, 1994. SPIE V.

[PCF*02a] PRINCE S., CHEOK A., FARBIZ F., WILLIAMSON T., JOHNSON N., BILLINGHURST M., KATO H.: 3D Live: Real time captured content for mixed reality. In *International Symposium on Mixed and Augmented Reality* (2002).

[PCF*02b] PRINCE S., CHEOK A. D., FARBIZ F., WILLIAMSON T., JOHNSON N., BILLINGHURST M., KATO H.: 3-D live: real time interaction for mixed reality. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work* (2002), ACM Press, pp. 364–371.

[PT02a] PIEKARSKI W., THOMAS B.: Arquake: the outdoor augmented reality gaming system. *Commun. ACM 45*, 1 (2002), 36–38.

[PT02b] PIEKARSKI W., THOMAS B. H.: The Tinmith system: demonstrating new techniques for mobile augmented reality modelling. In *Third Australasian conference on User Interfaces* (2002), Australian Computer Society, Inc., pp. 61–70.

[PT03] PIEKARSKI W., THOMAS B. H.: Interactive augmented reality techniques for construction at a distance of 3d geometry. In *Proceedings of the workshop on Virtual Environments 2003* (2003), ACM Press, pp. 19–28.

[PTB*01] POUPYREV I., TAN D. S., BILLINGHURST M., KATO H., REGENBRECHT H., TETSUTANI N.: Tiles: A mixed reality authoring interface. In *Proceedings of INTERACT 2001 Conference on Human Computer Interaction* (2001).

[RC03] ROMERO L., CORREIA N.: HyperReal: A hypermedia model for mixed reality. In *Proceedings of the fourteenth ACM conference on Hypertext and Hypermedia* (2003), ACM Press, pp. 2–9.

[SHM*02] STAPLETON C. B., HUGHES C. E., MOSHELL J. M., MICIKEVICIUS P., ALTMAN M.: Applying mixed reality to entertainment. *IEEE Computer 35*, 12 (2002), 122–124.

[SN04] STEINMETZ R., NAHRSTEDT K.: *Multimedia Applications*. Springer, Jan. 2004.

[ZHHL03] ZAUNER J., HALLER M., HARTMANN W., LUCKENEDER T.: *A generic framework for a training application based on Mixed Reality*. Tech. rep., MTD - Upper Austria University of Applied Sciences, Hagenberg, Austria, 2003.