CONTENT MANAGEMENT IN MIXED REALITY SYSTEMS

Considerations from an applied science perspective

Martin Kurze

T-Systems Nova, Berkom

kurze@berkom.de

Abstract: Content Management Systems form the backbone of most of today's interactive multimedia applications. Mixed Reality Systems are probably today's most ambitious multimedia applications. This paper discusses some consequences of these findings. A state of the art CMS architecture is described briefly, and a similar architecture is proposed for a general type of MR application. As a first result a considerable increase in complexity is observed from a software architects point of view. On the other hand, a structured way to deal with the various types of content should lead to better system quality. The paper does not present a final conclusion, but rather offers a number of key issues to consider in future MR applications utilizing content management systems.

Key words: content management, user interface, industrial application, MPEG-7

1. Introduction

Today, most multimedia applications make use of databases and presentation interfaces called content management systems (CMS). In particular web based applications edited by professional authors heavily rely on these CMSes. The more advanced the presented content is, the less off-the-shelf-systems exist to perform the CMS-task. One crucial aspect of advanced applications is interactivity; the other one is unstructured content (e.g. raw images or movies).

While currently the highest degree of interactivity is gained in Virtual Reality (VR) Systems, the least structured data available are live video streams. For both sorts of content, no satisfying tool for content management

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35660-0_65](https://doi.org/10.1007/978-0-387-35660-0_65)

R. Nakatsu et al. (eds.), *Entertainment Computing*

© IFIP International Federation for Information Processing 2003

is available. And now we are going to combine live video and audio with VR technology in Mixed Reality (MR) systems. Undoubtedly, Mixed Reality is one of the biggest challenges for content management approaches:

Part of the “content” stems from highly interactive VR-Systems and thus is hard to control (in particular since the user himself plays a central role in this play). The other part of the content is reality itself; this would result in a kind of “reality management tool” instead of a conventional content management tool.

So the problem scope is (more or less) clear. Let’s deal with the problem itself and sketch some solutions:

2. What is “content” in MR systems?

Mixed Reality systems today take input from four sources: the environment, a database, third party sources and user input (see also [1]).

From the *environment*, information gained with sensors for certain aspects of the environment. More exact: these sensors try to mimic human senses in order to produce stimuli for human senses later on. In typical implementations these sensors are a camera (optionally two or more cameras) and a microphone (optionally stereo).

From *databases* information is retrieved according to the current state of the user in the system. This content can be divided into several classes, regarding the origin of the information (other classifications are of course possible, but in our context we can restrict the focus on the classification by origin). Important parts of the database content come from sensors like those mentioned above: An MR system needs some knowledge about the real environment to integrate the augmentation into this environment. This knowledge is clearly some sort of “content”. In many systems this sort of information is by far the biggest amount of data stored. Peculiarly this information will almost never be shown to the user, it is used only internally to determine what the user would see if he would not use the MR system.

Other information stored in the database comes from human input, very often directly from the system designers. These data are meant to augment or complement the reality as it is displayed to the user during the MR session, e.g. planned buildings or agents.

Such information can be regarded as coming from *third party sources*: In the context of MR applications the system designers themselves provide today almost all the information in the system. In the (near) future commercial applications will require that certain aspects of the artificial part of the environment should be sponsored or delivered by some sort of customer. A very simple example for this kind of third party provided

content could be a banner to be displayed on a real wall or on a virtual building (ore elsewhere in the human computer interface).

Also abstract live information such as weather news, directional hints (for navigation), text-chat and so on are presented to the user. This sort of “content” is in most cases not stored in the database but nevertheless is transferred to the user and thus can be regarded as content, very often the most valuable content in commercial systems.

The last type of input comes from the *user* directly. This input can be a sequence of navigation acts. These do not change the rest of the content (except in multi user systems where each user is part of the environment of the other users). User input can also change the state of the environment, either the “real” part of it or the artificial. Both cases must be covered by the MR system, in particular by the database storing content information.

We can conclude this section by stating that the database for storing all the information about the real environment and the augmentation plays a central role in MR systems

3. What is content management?

Content management systems deal with three main objectives/tasks: Content acquisition, content storage and content presentation [3]. In multimedia systems, these tasks are often performed in two different layers, the data layer and the meta data layer. Multimedia data (as they are used un VR and MR systems) are more or less unstructured; examples are audio- and video files, images and geometry information. Meta data contains information about this multimedia data. It is in most cases very well structured; state of the art in Multimedia metadata formats is MPEG 7. One key issue for meta data acquisition is extracting structured information from unstructured data. MR systems do a very good job in this aspect: image analysis and other methods generate geometry information in near real time from the camera image from the user's headset-camera. Other structured metadata must be generated manually, such as names of actors or buildings, meta information about purely artificial components etc.

CMSes, in particular multimedia CMSes store unstructured data in file systems or database-BLOBs (Binary Large Objects). Structured meta data can be stored in more elaborate formats, preferably in MPEG 7. MPEG 7 uses XML to structure multimedia metadata. While MPEG 7 claims to store high level and low level metadata and thus to allow advanced search and retrieval functions, today only high level data is taken into consideration. However, content (data) and metadata are linked by pointers in the MPEG 7 data sets. Due to the XML nature of metadata (in MPEG 7) specialized

database systems should be used; traditional RDBMSes often fall short here.

Content retrieval is one of the key functionalities in CMS; however, in MR systems content retrieval is in most cases done implicitly: the user navigates and acts in Mixed Reality, and the system retrieves the data needed based on the user's action. Nevertheless, explicit search and retrieval happens if the user decides to request information or if the user watches a movie or an image in the MR environment. This type of content is often inserted, stored and managed conventionally in the CMS.

The last aspect of content management is content presentation. Again we have to differentiate between implicit and explicit presentation: A large portion of the content in MR-CMSes is not displayed; rather it is used to control the presentation of the artificial components. All the presented contents from the CMS in MR are piped through a rendering unit and integrated in the user's view. This results in certain requirements for the performance of the CMS and in interface needs for transporting data from the CMS to the renderer.

Figure 1 shows a traditional and well-understood architecture for a content management system. The following paragraphs briefly describe the roles of the individual components, modules and persons in the system.

The whole system is split vertically in two modules: The Editing System forms the backend of the CMS, and the Publishing System forms the front-end including the end user interface.

The Editing System is where the main database is located. This main database is then divided into two layers: the metadata layer contains the information about the data, e.g. creator's information, copyright information and information about the period of time to use a particular piece of data in the system. The actual data is most often represented as text (HTML, XML or a similar format). This text contains rules about the structure of the data and links to (multi)media assets.

Media assets are generally stored outside the core database because they are only used and referenced by structured textual datasets (which are stored in the database). Database searches are not performed on the media data but rather on the media-*meta* data and within the structured text in the database (see [2] for details).

One interesting media type should be mentioned at this point: live streams. Live audio or video streams are clearly multimedia content. But this content is not known in advance, and thus these streams cannot be treated like regular media assets. No content quality assurance is applicable. Also the stream source is generally outside of the CMS's scope; only a link is maintained within the CMS.

Within the Editing System two main roles can be identified (besides the technical administrators who keep the system running): Editors create and update contents (text, structure and media contents), and content providers

deliver preformatted pieces of content, mainly media and text content. The main difference between these roles is the rights-schema: An editor can change almost everything in the system while a content provider can only add and remove contents.

Figure 1. Typical Architecture of a content management system

The Publishing System contains a partial copy of the CMS database. Most of the metadata is left behind, but the complete set of structured data is stored in this module. In most cases the media assets are not copied but referenced via links from the Editing System and the Publishing System.

The important step to copy data from the Editing System to the Publishing System is controlled by one person/role who is finally responsible for the quality of the (published) contents. Both modules (Editing and Publishing System) include a reporting mechanism to allow customers, sponsors and lawyers to check what actually happened in the system (who watched/inserted/changed which piece of content when).

The last component in our sample CMS is the user interface for the end user, e.g. a customer or web surfer. In regular systems this user interface is realized as a combination of a web server and a client such as MS Internet

Explorer, Netscape navigator or Opera running an html rendering engine including some plug-ins for the media assets. This interface is more or less a presentation interface and allows only very limited feedback to the CMS.

And this is where an MR CMS enters the stage.

4. Specific requirements for CMS in MR applications

Figure 2 shows a proposed architecture for a CMS for MR applications based on the standard architecture described above. Most changes are highlighted in bold in the figure.

Figure 2. Proposed architecture for a CMS in MR

The primary change to be observed in this architecture compared to figure 1 is the role and weight of the interface. Actually the interface is not any longer just a part of the publishing system presenting some content to the user. Rather the user resides inside the interface, which in turn is not only a part of the system but at the same time part of the environment. The user acts inside of the interface. This means also that virtually every action

performed by the user can be considered as system input. Whatever the user does is considered to be relevant for the environment or the system state. This is the typical situation in AR/MR systems.

This high degree of immersion results in a number of new arrows in the figure, i.e. new effects in the CMS-centred view to the MR application:

First, the user changes the environment with every action. In the simplest case, this change can be just a change of the users position in the environment which by definition changes the environment (the user is a part of the environment); or this change can be an actual manipulation in the environment, e.g. opening a door or rotating a real object.

Then, the user is observed by the sensors of the system, e.g. external cameras or body trackers. These changes need to be recorded and evaluated by the MR-application because they might have a meaning and/or an effect to the virtual part of the environment.

The evaluation of these input data will most likely cause changes in the systems meta-model of the whole situation/world. In our CMS-view the meta model is stored in the metadata database in the editing system.

Finally, the user himself influences the video streams transferred to the user. In the case of video-overlay display (for the user), the helmet camera is directly controlled by the user's head movements and thus cannot be managed by the content management system.

Summarizing the observations, we can state that the user in MR based CMSes is not just an information/content retriever anymore but also rather a part time content editor. The consequences for the publishing process and other aspects will be discussed below.

5. Why all the Hassle?

As we have seen above, CMSes solve a few problems of today's MR systems and add several new ones. So why should anyone use a CMS? After all, MR systems work quite good today without them. The answer lies in the question. Today MR systems are not very large and are often not built with commercial use in mind. Tomorrow all those sponsors who financed basic research until today will require commercialisation. And tomorrow MR systems will be much larger than those, which we use today. So tomorrow there will be no question whether we need CMS or not. Tomorrow the question will be: How could we live in MR without CMS? Then we have to be prepared to answer questions towards the methods used to manage huge amounts of structured and unstructured data from various sources with various intellectual property rights attached to them.

Talking about rights brings us to the last aspect of this paper:

commercialisation. How can we make money with MR systems? We cannot sell reality. Reality is still free, even free of virtual enhancements. But we can sell the knowledge about reality, which is used in MR systems: The quality of the knowledge can directly affect the price for the customer.

In addition, all the extra data in MR systems can be charged for. Guidance can be accounted; extra entertainment (e.g. more or better agents in MR) can be billed.

Only advanced and to a large degree still to be developed systems can fulfil the requirements sketched in this paper. The MR community should see this as a chance to make MR in entertainment a success in pure (commercial) reality.

6. Conclusions and Perspectives

The changes and special features of MR adjusted content management systems almost seem to break the methodology of a clean conventional CMS:

The CMS has to deal with the environment as a whole; ultimately it might need “world knowledge” much in the way AI systems utilize it.

The CMS has to treat the user as a perceiver of content and as an editor or provider of content. The role of a content provider seems to be in several aspects more appropriate than the role of an editor.

The CMS has to treat real-time information (in both directions, input and output) in special ways to overcome the bottleneck of quality control by some instance such as a chief editor. This should not result in lower quality of the content!

For the future it seems to be advisable to define the role of an MR-user in MR related CMSes. The CMS approach already proved to increase the economical success of multimedia applications and may help MR systems to conquer the mass market. This requires a new style of co-operation between database and publishing experts and the VR/MR community. Such a joint efforts often created unexpected and interesting results.

7. References

- [1] Yuichi Ohta & Hideyuki Tamura, *Mixed Reality – Merging Real and Virtual Worlds*, Berlin: Springer, 1999.
- [2] Peter Schäuble, *Multimedia Information Retrieval: Content-Based Information Retrieval from Large Text and Audio Databases*, Boston: Kluwer, 1997.
- [3] Bob Boiko, *Content Management Bible*, New York: Hungry Minds Inc, 2001