A. Scherp and S. Boll: *MM4U - A framework for creating personalized multimedia content*, In: Managing Multimedia Semantics, S. Nepal and U. Srinivasan (Ed.), Chapter 11, pages 246-287, Idea Group Publishing, 2005. ISBN 1-59140-542-4

MM4U: A Framework for Creating Personalized Multimedia Content

Ansgar Scherp, OFFIS Research Institute, Germany
Susanne Boll, University of Oldenburg, Germany

Abstract: *In the Internet age and with the advent of digital multimedia information, we succumb the possibilities that the enchanting multimedia information seems to offer but end up in almost drowning in the multimedia information: Too much information at the same time, so much information that is not suitable for the current situation of the user, too much time needed to find information that is really helpful. The multimedia material is there, but the issue how the multimedia content is found, selected, assembled, and delivered such that it is most suitable for the user's interest and background, the user's preferred device, network connection, location, and many other settings is far away from being solved.*

In this chapter, we are focusing on the aspect of how to assemble and deliver personalized multimedia content to the users. We present the requirements and solutions of multimedia content modeling and multimedia content authoring as we find it today. Looking at the specific demands of creating personalized multimedia content we come to the conclusion that a dynamic authoring process is needed in which just in time the individual multimedia content is created for a specific user or user group.

We designed and implemented an extensible software framework, MM4U, which provides generic functionality for typical tasks of a dynamic multimedia content personalization process. Such a framework at hand, an application developer can concentrate on creating personalized content in the specific domain and at the same time is relieved from the basic task of selecting, assembling, and delivering personalized multimedia content. We present the design of the MM4U framework in detail with an emphasis for the personalized multimedia composition and illustrate the framework's usage in the context of our prototypical applications.

Keywords: Multimedia, Multimedia Application, Multimedia Component Architecture, Hypermedia, Hypertext, Storyboard

1. INTRODUCTION

Multimedia content today can be considered as the composition of different media elements such as images and text, audio, and video into an interactive multimedia presentation like a guided tour through our home town Oldenburg. Features of such a presentation are typically the temporal arrangement of the media elements in the course of the presentation, the layout of the presentation, and its interaction features. *Personalization* of multimedia content means that the multimedia content is targeted at a specific person and reflects this person's individual context, specific background, interest, and knowledge, as well as the heterogeneous infrastructure of end devices to which the content is delivered and on which it is presented. The creation of personalized multimedia content means that for each intended context a custom presentation needs to be created. Hence, multimedia content personalization is the shift from one-size-fits-all to a very individual and personal one-to-one provision of multimedia content to the users. This means in the end that the multimedia content needs to be prepared for each individual user. However, if there are many different users that find themselves in very different contexts it becomes soon obvious that a manual creation of different content for all the different user contexts is not feasible let alone economical (see also André et al., 1996). Rather a *dynamic*, (automated) process of selecting and assembling personalized multimedia content depending on the user context seems to be reasonable.

The creation of multimedia content is typically subsumed under the notion of multimedia authoring. However, such authoring today is seen as the static creation of multimedia content. Authoring tools with graphical user interfaces allow to manually create content that is targeted at a specific user group. If the content created is at all “personalizable”, then only within a very limited scope. First research approaches in the field of dynamic creation of personalized multimedia content are promising; however, they are often limited to certain aspects of the content personalization to the individual user. Especially, whenever the content personalization task is more complex, these systems need to employ additional programming. As we observe that programming is needed in many cases anyway, we continue this observation consequently and propose with MM4U (short for “MultiMedia for you”) a component-based object-oriented software framework to support the software development process of multimedia content personalization applications. MM4U relieves application developers from general tasks in the context of multimedia content personalization and lets them concentrate on the application domain-specific tasks. The framework's components provide generic functionality for typical tasks of the multimedia content personalization process. The design of the framework is based on a comprehensive analysis of the related approaches in the field of user profile modeling, media data modeling, multimedia composition, and multimedia presentation formats. We identify the different tasks that arise in the context of creating personalized multimedia content. The different components of the framework support these different tasks for creating user-centric multimedia content: They integrate the generic access to user profiles, media data, and associated meta data, provide support for personalized multimedia composition and layout, as well as the creation of the context-aware multimedia presentations. With such a framework the development of multimedia applications becomes easier and much more efficient for different users with their different (semantic) contexts. On the basis of the MM4U framework, we are currently developing two sample applications, a personalized multimedia sightseeing tour and a personalized multimedia sports news ticker. The experiences we gain from the development of these applications give us important feedback on the evaluation and continuous re-design of the framework.

The remainder of this chapter is organized as follows: To review the notion of multimedia content authoring, in Section 2 we present the requirements of multimedia content modeling and the authoring support we find today. Setting off from this, Section 3 introduces the reader into the tasks of creating personalized multimedia content and why such content can be created only in a dynamic fashion. In Section 4, we address the related approaches we find in the field before we present the design of our MM4U framework in Section 5. As the personalized creation of multimedia content is a central aspect of the framework, Section 6 presents in detail the multimedia personalization features of the framework. Section 7 shows how the framework supports application developers and multimedia authors in their effort to create personalized multimedia content. The implementation and first prototypes are presented in Section 8 before we come to our summary and conclusion in Section 9.

2. MULTIMEDIA CONTENT AUTHORTING TODAY

In this section, we introduce the reader in current notions and techniques of multimedia content modeling and multimedia content authoring. An understanding of requirements and approaches in modeling and authoring of multimedia content is a helpful prerequisite to our goal, the dynamic creation of multimedia content. For the modeling of multimedia content we present our notion of multimedia content, documents, and presentation and describe the central characteristics of typical multimedia document models in Section 2.1. For the creation of multimedia content, we give a short overview of directions in multimedia content authoring today in Section 2.2.

2.1. Multimedia content

Multimedia content today is seen as the result of a composition of different media elements (media content) in a continuous and interactive multimedia presentation. Multimedia content builds on the modeling and representation of the different media elements that form the building bricks of the composition. A multimedia document represents the composition of continuous and discrete media elements into a logically coherent multimedia unit. A multimedia document that is composed in advance to its rendering is called pre-orchestrated in contrast to compositions that take place just before rendering that are called live or *on-the-fly*. A multimedia document is an instantiation of a *multimedia document model* that provides the primitives to capture all aspects of a multimedia document. The power of the multimedia document model determines the degree of the multimedia functionality that documents following the model can provide. Representatives of (abstract) multimedia document models in research can be found with CMIF (Bulterman et al., 1991), Madeus (Jourdan et al., 1998), Amsterdam Hypermedia Model (Hardman 1998; Hardman et al., 1994), and ZYX (Boll & Klas, 2001). A *multimedia document format* or *multimedia presentation format* determines the representation of a multimedia document for the document's exchange and rendering. Since every multimedia presentation format implicitly or explicitly follows a multimedia document model, it can also be seen as a proper mean to "serialize" the multimedia document's representation for the purpose of exchange. Multimedia presentation formats can either be standardized such as the W3C standard SMIL (Ayars et al., 2001) or proprietary such as the wide spread Shockwave file format (SWF) of Macromedia (Macromedia, 2004). A *multimedia presentation* is the rendering of a multimedia document. It comprises the continuous rendering of the document in the target environment, the (pre)loading of media data, realizing the temporal course, the temporal synchronization between continuous media streams, the adaptation to different or changing presentation conditions and the interaction with the user.

Looking at the different models and formats we find and also the terminology in the related work there is not necessarily a clear distinction between multimedia documents models and multimedia presentation formats as well as between multimedia documents and multimedia presentations. In this chapter, we distinguish the notion of multimedia document models as the definition of the abstract composition capabilities of the model, a multimedia document as an instance of this model. The term *multimedia content* or *content representation* is used to abstract from existing formats and models and generally addresses the composition of different media elements into a coherent multimedia presentation. Independent of the actual document model or format chosen for the content, one can say that a multimedia content representation has to realize at least three central aspects – the temporal, spatial, and interactive characteristics of a multimedia presentation (Boll et al., 2000). However, as many of today's concrete multimedia presentation formats can be seen as representing both a document model and an exchange format for the final rendering of the document we use these as an illustration of the central aspects of multimedia documents. We present an overview of these characteristics in the following; for a more detailed discussion on the characteristics of multimedia document models we refer the reader to (Boll et al., 2000; Boll & Klas, 2001).

- A *temporal model* describes the temporal dependencies between media elements of a multimedia document. With the temporal model, the temporal course such as the parallel presentation of two videos or the end of a video presentation on a mouse click event can be described. One can find four types of temporal models: *point-based* temporal models, *interval-based* temporal models (Little & Ghafoor, 1993; Allen, 1983), *enhanced interval-based* temporal models that can handle time intervals of unknown duration (Duda & Keramane, 1995; Hirzalla et al., 1995; Wahl & Rothermel, 1994), *event-based* temporal models, and *script-based* realization of temporal relations. The multimedia presentation formats we find today realize different temporal models, e.g., SMIL 1.0 (Bugaj et al., 1998) provides an interval-based temporal model only, while SMIL 2.0 (Ayars et al., 2001) also supports an event-based model.
- For a multimedia document not only the temporal synchronization of these elements is of interest but also their spatial positioning on the presentation media, e.g., a window, and possibly the spatial relationship to other visual media elements. The positioning of visual media element in the multimedia presentation can be expressed by the use of a *spatial model*. With it one can, for example, place one image about a caption or define the overlapping of two visual media. Besides the arrangement of media elements in the presentation also the visual layout or design is defined in the presentation. This can range from a simple setting for background colors and fonts up to complex visual design and effects. In general, three approaches to spatial models can be distinguished: *absolute positioning*, *directional relations* (Papadias et al., 1995; Papadias & Sellis, 1994), and *topological relations* (Egenhofer & Franzosa, 1991). With absolute positioning we subsume both the placement of a media element at an absolute position with respect to the origin of the coordinate system and the placement at an absolute position relative to another media element. The absolute positioning of media elements can be found, e.g., with Flash (Macromedia, 2004) and the Basic Language Profile of SMIL 2.0, while the relative positioning is realized, e.g., by SMIL 2.0 and SVG 1.2 (Andersson et al., 2004b).
- A very distinct feature of a multimedia document model is the ability to specify *user interaction* in order to let a user choose between different presentation paths. Multimedia documents without user interaction are not very interesting as the course of their presentation is exactly known in advance and, hence, could be recorded as a movie. With interaction models a user can, for example, select or repeat parts of presentations, speed up a movie presentation, or change the visual appearance. For the modeling of user interaction, one can identify at least three basic types of interaction: *navigational interactions*, *design interactions*, and *movie interactions*. Navigational interaction allows the selection of one out of many presentation paths and is supported by all the considered multimedia document models and presentation formats.

Looking at existing multimedia document models and presentation formats both in industry and research, one can see that these aspects of multimedia content are implemented in two general ways: The standardized formats and research models typically implement these aspects in different variants in a structured (XML) fashion as can be found with SMIL 2.0, HTML+TIME (Schmitz et al., 1998), SVG 1.2, Madeus, and ZYX. Proprietary approaches, however, represent or program these aspects in an adequate internal model such as Macromedia's Shockwave format. Independent of the actual multimedia document model, support for the *creation* of these documents is needed – multimedia content authoring. We will look at the approaches we find in the field of multimedia content authoring in the next section.

2.2. Multimedia authoring

While multimedia content represents the composition of different media elements into a coherent multimedia presentation, *multimedia content authoring* is the process in which this presentation is actually created. This process involves parties from different fields including media designers, computer scientists, and domain experts: Experts from the domain provide their knowledge in the field; this knowledge forms the input for the creation of a storyboard for the intended presentation. Such a storyboard forms often the basis on which creators and directors plan the implementation of the story with the respective media and with which writers, photographers, and camerapersons acquire the digital media content. Media designers edit and process the content for the targeted presentation. Finally, multimedia authors compose the preprocessed and prepared material into the final multimedia presentation. Even though we described this as a sequence of steps, the authoring process typically includes cycles. In addition, the expertise for some of the different tasks in the process can also be hold by one single person. In this paper, we are focusing on the part of the multimedia content creation process in which the prepared material is actually assembled into the final multimedia presentation.

This part is typically supported by professional multimedia development programs, so-called authoring tools or authoring software. Such tools allow the composition of media elements into an interactive multimedia presentation via a graphical user interface. The authoring tools we find here range from domain expert tools to general purpose authoring tools.

- Domain expert tools hide as much as possible the technical details of content authoring from the authors and let them concentrate on the actual creation of the multimedia content. The tools we find here are typically very specialized and targeted at a very specific domain. An example for such a tool has been developed in the context of our previous research project Cardio-OP (Klas et al., 1999) in the domain of cardiac surgery. The content created in this project is an interactive multimedia book about topics in the specialized domain of cardiac surgery. Within the project context, an easy-to-use authoring wizard was developed to allow medical doctors to easily create “pages” of a multimedia book in cardiac surgery. The Cardio-OP-Wizard guides the domain experts through the authoring process by a digital storyboard for a multimedia book on cardiac surgery. The wizard hides thereby as much technical details as possible.
- On the other end of the spectrum of authoring tools we find highly generalized tools such as Macromedia Director (Macromedia, 2004). These tools are independent of the domain of the intended presentation and let the authors create very sophisticated multimedia presentations. However, the authors typically need to have high expertise in using the tool. Very often programming in an integrated programming language is needed to achieve special effects or interaction patterns. Consequently, the multimedia authors need programming skills and along with this some experience in software development and software engineering.

Whereas a multimedia document model has to represent the different aspects of time, space, and interaction, multimedia authoring tools must allow the authors to actually assemble the multimedia content. However, the authors are normally experts from a specific domain. Consequently, only those authoring tools are practicable to create multimedia content for a specific domain, which are highly specialized and easy to use.

3. DYNAMIC AUTHORING OF PERSONALIZED CONTENT

The authoring process described above so far represents a manual authoring of multimedia content, often with high effort and cost involved. Typically, the result is a multimedia presentation targeted at a certain user group in a special technical context. However, the one-size-fits-all fashion of the multimedia content created does not necessarily satisfy the different users' needs. Different users may have different preferences concerning the content and also may access the content in networks on different end devices. For a wider applicability the authored multimedia content needs to "carry" some alternatives that can be exploited to adapt the presentation to the specific preferences of the users and their technical settings. Figure 1 shows an illustration of the variation possibilities that a simple personalized city guide application can possess. The root of the tree represents the multimedia presentation for the personalized city tour. If this presentation was intended for both Desktop PC and PDA, this results in two variants of the presentation. If then some tourists are interested only in churches, museums, or palaces and would like to receive the content in either English or German this already sums up to 12 variants. If then the multimedia content should be available in different presentation formats the number of variation possibilities within a personalized city tour increases again. Even though different variants are not necessarily entirely different and may have overlapping content the example is intended to illustrate that the flexibility of multimedia content to personalize to different user contexts quickly leads to an explosion of different options. And still the content can only be personalized within the flexibility range that has been anchored in the content.

From our point of view, an efficient and competitive creation of personalized multimedia content can only result in a system approach that supports the *dynamic* authoring of personalized multimedia content. A dynamic creation of such content allows for a selection and composition of just those media elements that are targeted at the user's specific interest and preferences. Generally, the dynamic authoring comprises the steps and tasks that occur also with static authoring, only with the difference that the creation process is postponed to the time when the targeted user context and the presentation is just created for this specific context. To be able to efficiently create presentations for (m)any given contexts a manual authoring of a presentation meeting the user is not an option, rather a dynamic content creation is needed.

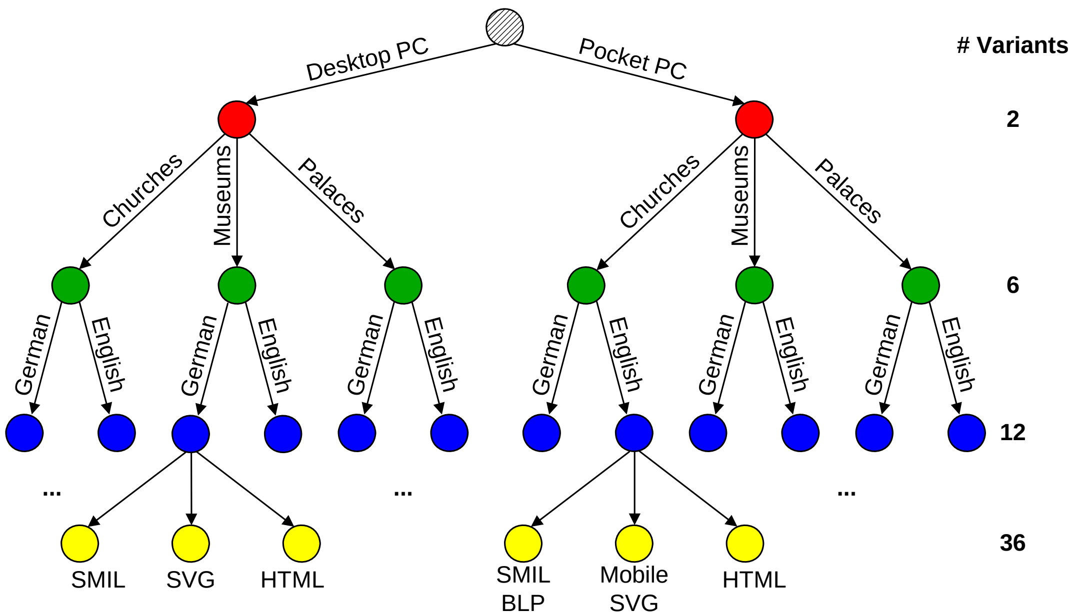

Figure 1: Example of the variation possibilities within a personalized city guide application

In the following, we look into the process of *dynamic authoring* of personalized multimedia content. This process involves different phases and tasks. We identify the central tasks in this process that need to be supported by a suitable solution for personalized content creation.

Figure 2 depicts the general process of creating personalized multimedia content. The core of this process is an application we call *personalization engine*. The input parameters to this engine can be characterized by three groups: The first group of input parameters is the media elements with the associated meta data that constitute the content from which the personalized multimedia presentations are selected and assembled. The second group enfolds the user's personal and technical context. The user profile includes information about, e.g., the user's current task, the location, and environment, like weather and loudness, his or her knowledge, goals, preferences and interests, abilities and disabilities, as well as demographic data. The technical context is described by the type of the user's end device, the hardware and software characteristics, as for example the available amount of memory and media player, as well as possible network connections and input devices. The third group of input parameters influences the general structure of the resulting personalized multimedia presentation and subsumes other preferences a user could have for the multimedia presentation.

Within the personalization engine, these input parameters are now used to author the personalized multimedia presentation. First, the personalization engine exploits all available information about the user's context and his or her end device to select by means of media meta data those media elements that are of most relevance according to the user's interests and preferences and meet the characteristics of the end device at the best. In the next step, the selected media elements are assembled and arranged by the personalization engine – again in regard of the user profile information and the characteristics of the end device – to the personalized multimedia content, represented in an *internal document model* (Scherp & Boll, 2004b). This internal document model abstracts from the different characteristics of today's multimedia presentation formats and, hence, forms the greatest common denominator of these formats. Even though our abstract model does not reflect the fancy features of some of today's multimedia presentation formats, it supports the very central multimedia features of modeling time, space, and interaction. It is designed to be efficiently transformed to the concrete syntax of the different presentation formats. For the assembly, the personalization engine uses the parameters for document structure, the layout and style parameters, and other rules and constraints that describe the structure of the personalized multimedia presentation, to determine among others the temporal course and spatial layout of the presentation. The center of Figure 2 sketches this temporal and spatial arrangement of selected media elements over time in a spatial layout following the document structure and other preferences. Only then in the transformation phase, the multimedia content in the internal document model is transformed to a concrete presentation format. Finally, the just generated personalized multimedia presentation is rendered and displayed by the actual end device.

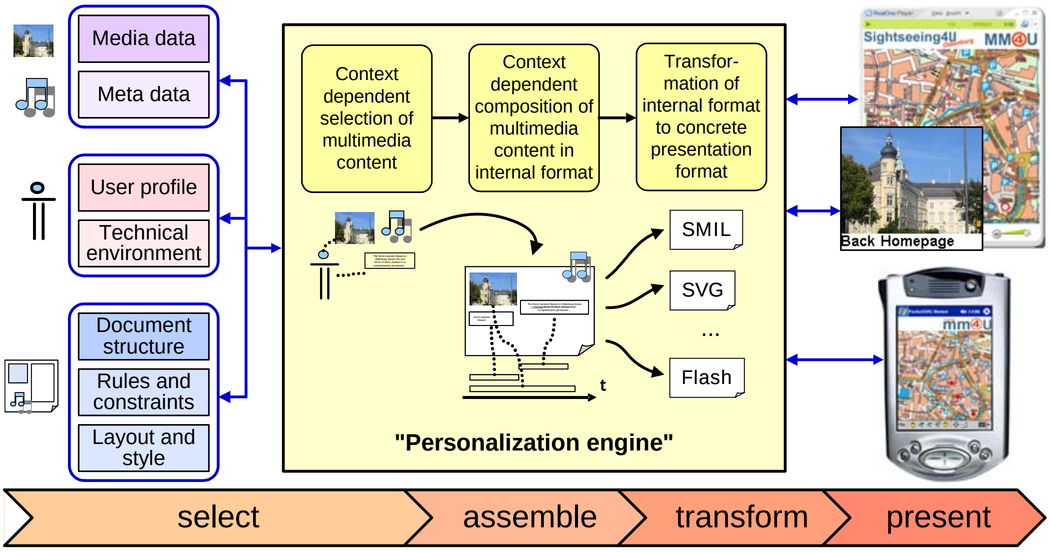

Figure 2: The general process of personalizing multimedia content

4. RELATED APPROACHES

In the following we present the related approaches in the field of personalized multimedia content creation. We first discuss the creation of personalizable multimedia content with today's authoring environments before we come to research approaches that address a *dynamic* composition of adapted or personalized multimedia content.

Multimedia authoring tools like Macromedia Director (Macromedia, 2004) today require high expertise from their users and do create multimedia presentations that are targeted only at a specific user or user group. Everything "personalizable" needs to be programmed or scripted within the tool's programming language. Early work in the field of creating advanced hypermedia and multimedia documents can be found, e.g., with the Amsterdam Hypermedia Model (Hardman 1998; Hardman et al., 1994b) and the authoring system CMIFed (van Rossum 1993; Hardman et al., 1994a) as well as with the ZYX (Boll & Klas, 2001) multimedia document model and a domain-specific authoring wizard (Klas et al., 1999). In the field of standardized models, the declarative description of multimedia documents with SMIL allows for the specification of adaptive multimedia presentations by defining presentation alternatives by using the "switch" element. A manual authoring of such documents that are adaptable to many different contexts is too complex; also the existing authoring tools such as GR*i*NS editor for SMIL from Oratrix (Oratrix, 2004) are still tedious to handle. Some SMIL tools provide support for the "switch" element to define presentation alternatives; a comfortable interface for editing the different alternatives for many different contexts, however, is not provided. Consequently, we have been working on the approach in which a multimedia document is authored for one general context and is then "automatically" enriched by the different presentation alternatives needed for the expected user contexts in which the document is to be viewed (Boll et al., 1999). However, this approach is reasonable only for a limited number of presentation alternatives and limited presentation complexity in general.

Approaches that *dynamically* create personalized content are typically found on the Web, e.g., Amazon.com (Amazon, 1996-2004) or MyYahoo (Yahoo!, 2002). However, these systems remain text-centric and are not occupied with the complex composition of media data in time and space into real multimedia presentations. On the pathway to an automatic generation of *personalized* multimedia presentations, we primarily find research approaches that address personalized *media* presentations only: For example, the home-video editor Hyper-Hitchcock

(Girgensohn, 2003; Girgensohn et al., 2001) provides a pre-processing of a video such that users can interactively select clips to create their personal video summary. Other approaches create summaries of music or video, e.g., (Kopf et al., 2004; Agnihotri et al., 2003). However, the systems provide an intelligent and intuitive access to large sets of (continuous) media rather than a dynamic creation of individualized content. An approach that addresses personalization for videos can be found, e.g., with IBM's Video Semantic Summarization System (IBM Corporation, 2004a) which is, however, still concentrating on one single media type.

Towards *personalized multi*media we find interesting work in the area of *adaptive hypermedia* systems which is going on for quite some years now (Brusilovsky 1996; Wu, 2001; De Bra et al., 1999a, 2000, 2002b; De Carolis et al., 1998, 1999). The adaptive hypermedia system AHA! (De Bra et al., 1999b; 2002a, 2003) is a prominent example here which also addresses the authoring aspect (Stash & De Bra, 2003), e.g., in adaptive educational hypermedia applications (Stash et al., 2004). However, though these and further approaches integrate media elements in their adaptive hypermedia presentations, synchronized multimedia presentations are not in their focus.

Personalized or adaptive *user interfaces* allow the navigation and access of information and services in a customized or personalized fashion. For example work done in the area of personalized agents and avatars considers "presentation generation" exploiting natural language generation and visual media elements to animate the agents and avatars (de Rosis et al., 1999). These approaches address the human computer interface; the general issue of dynamically creating arbitrary personalized multimedia content that meets the user's information needs is not in their research focus.

A very early approach towards the dynamic creation of multimedia content is the Coordinated Multimedia Explanation Testbed (COMET), which bases on an expert-system and different knowledge databases and uses constraints and plans to actually generate the multimedia presentations (Elhadad et al., 1991; McKeown et al., 1993). Another interesting approach to automate the multimedia authoring process has been developed at the DFKI in Germany by the two knowledge-based systems WIP (Knowledge-based Presentation of Information) and PPP (Personalized Plan-based Presenter). WIP is a knowledge-based presentation system that automatically generates instructions for the maintenance of technical devices by plan generation and constraint solving. PPP enhances this system by providing a life-like character to present the multimedia content and by considering the temporal order in which a user processes a presentation (André et al., 1996; André & Rist, 1995+1996). Also a very interesting research approach towards the dynamic generation of multimedia presentations is the Cuypers system (van Ossenbruggen et al., 2000) developed at the CWI. This system employs constraints for the description of the intended multimedia programming and logic programming for the generation of a multimedia document (CWI, 2004). The multimedia document group at INRIA in France developed within the Opéra project a generic architecture for the automated construction of multimedia presentations based on transformation sheets and constraints (Villard, 2001). This work is continued within the succeeding project Web, Accessibility, and Multimedia (WAM) with the focus on a negotiation and adaptation architecture for multimedia services for mobile devices (Lemlouma & Layaïda, 2003+2004).

However, we find limitations with existing systems when it comes to their expressiveness and flexible personalized content creation support. Many approaches for personalization are targeted at a specific application domain in which they provide a very specific content personalization task. The existing research solutions typically use a declarative description like rules, constraints, style sheets, configuration files, and the like to express the dynamic, personalized multimedia content creation. However, they can solve only those presentation generation problems that can be covered by such a declarative approach; whenever a complex and application-specific personalization generation task is required, the systems find their limit and need additional programming to solve the problem. Additionally, the approaches we find usually rely on fixed data models for describing user profiles, structural presentation constraints, technical infrastructure, rhetorical structure, etc. and use these data models as an input to their personalization engine. The latter evaluates the input data, retrieves the most suitable content, and tries to most intelligently compose the media into a coherent aesthetic multimedia presentation. A change of the input data models as well as an adaptation of the presentation generator to more complex presentation generation tasks is difficult if not unfeasible. Additionally, for these approaches the border between the declarative descriptions for describing content personalization constraints and the additional programming needed is not clear and differs from solution to solution. This leads us to the development of a software framework that supports the development of personalized multimedia applications.

5. THE MULTIMEDIA PERSONALIZATION FRAMEWORK

Most of the research approaches presented above apply to text-centered information only, are limited with regard to the "personalizability", or are targeted at very specific application domains. As mentioned above, we find that existing research solutions in the field of multimedia content personalization provide interesting solutions. They typically use a declarative description like style sheets, transformation rules, presentation constraints, configuration files, and the like to express the dynamic, personalized multimedia content creation. However, they can solve only those presentation generation problems that can be covered by such a declarative approach; whenever a complex and application-specific personalization generation task is required, the systems find their limit and need additional programming to solve the problem. To provide application developers with a general, domain independent support for the creation of personalized multimedia content we pursue a software engineering approach: the MM4U framework. With this framework, we propose a component-based object-oriented software framework that relieves application developers from general tasks in the context of multimedia content personalization and lets them concentrate on the application domain-specific tasks. It supports the dynamic generation of arbitrary personalized multimedia presentations and therewith provides substantial support for the development of personalized multimedia applications. The framework does not re-invent multimedia content creation but incorporates existing research in the field and also allows to be extended by domain and application-specific solutions. In the following Section 5.1, we identify by an extensive study of related work and own experiences the general design goals of this framework. In Section 5.2 we present the general design of the MM4U framework. Finally, a detailed insight into the framework's layered architecture is given in Section 5.3.

5.1. General design goals for the MM4U framework

The overall goal of MM4U is to simplify and to cheapen the development process of personalized multimedia applications. Therefore, the MM4U framework has to provide the developers with support for the different tasks of the multimedia personalization process as shown in Figure 2. These tasks comprise assistance for the access to media data and associated meta data as well as user profile information and the technical characteristics of the end device. The framework must also provide for the selection and composition of media elements into a coherent multimedia presentation. Finally, the personalized multimedia content must be created for delivery and rendering on the user's end device.

In regard of these different tasks, we conducted an extensive study of related work: In the area of user profile modelling we considered among others Composite Capability/Preference Profile (Klyne, 2003), FIPA Device Ontology Specification (Foundation for Intelligent Physical Agents, 2002), User Agent Profile (Open Mobile Alliance, 2003), Customer Profile

Exchange (Bohrer & Holland, 2004), (Fink et al., 1997), and (Chen & Kotz, 2000). In regard to meta data modeling, we studied different approaches of modelling meta data and approaches for meta data standards for multimedia, e.g., Dublin Core (Dublin Core Metadata Initiative, 1995-2003) and Dublin Core Extensions for Multimedia Objects (Hunter, 1999), Resource Description Framework (Beckett & McBride 2003), and the MPEG-7 Multimedia content description standard (ISO/IEC JTC 1/SC 29/WG 11, 1999, 2001a-e). For multimedia composition we analyzed the features of multimedia document models like for example SMIL (Ayars et al., 2001), SVG (Andersson et al., 2004b), Macromedia Flash (Macromedia, 2004), Madeus (Jourdan, 1998), and ZYX (Boll & Klas, 2001). For the presentation of multimedia content respective multimedia presentation frameworks were regarded like Java Media Framework (Sun Microsystems, 2004), MET++ (Ackermann 1996), and PREMO (Duke et al., 1999). Furthermore, other existing systems and general approaches for creating personalized multimedia content were considered as for example the Cuypers engine (van Ossenbruggen et al., 2000) and the Standard Reference Model for Intelligent Multimedia Presentation Systems (Bordegoni et al., 1997).

We also derived design requirements to the framework from first prototypes of personalized multimedia applications we developed in different fields such as a personalized sightseeing tour through Vienna (Boll, 2003), a personalized mobile paper chase game (Boll et al., 2003), and a personalized multimedia music newsletter.

From the extensive study of related work and the first experiences and requirements we gained from our prototypical applications we developed the single layers of the framework. We also derived three general design goals for MM4U. These design goals are:

- The framework is to be designed such that it is independent of any special application domain, i.e., it can be used to generate arbitrary personalized multimedia content. Therefore, it provides general multimedia composition and personalization functionality and is flexible enough to be adapted and extended concerning the particular requirements of the concrete personalization functionalities a personalized application needs.
- The access to user profile information and media data with its associated meta data must be independent of the particular solutions for storage, retrieval, and processing of such data. Rather the framework should provide a unified interface for the access to existing solutions. With distinct interfaces for the access to user profile information and media data with associated meta data, it is the framework's task to use and exploit existing (research) profile and media storage systems for the personalized multimedia content creation.
- The third design goal for the framework is what we call presentation independence. The framework is to be independent of, e.g., the technical characteristics of the end devices, their network connection, and the different multimedia output formats that are available. This means, that the framework can be used to generate equivalent multimedia content for the different users and output channels and their individual characteristics. This *multi channel usage* implies that the personalized multimedia content generation task is to be partitioned into a composition of the multimedia content in an internal *representation format* and its later transformation into arbitrary (preferably standardized) *presentation formats* that can be rendered and displayed by end devices.

These general design goals have a crucial impact on the structure of the multimedia personalization framework, which we present in the following section.

5.2. General design of the MM4U framework

A software framework like MM4U is a semifinished software architecture, providing a software system as a generic application for a specific domain (Pree, 1995). The MM4U framework comprises components, which are bound together by their interaction (Szyperski et al., 2002), and realizes generic support for personalized multimedia applications. Each component is realized as an object-oriented framework and consists of a set of abstract and concrete classes. Depending on the usage of a framework so-called "white-box" and "black-box" frameworks can be distinguished (respectively white-box and gray-box reuse). A framework is used as a black-box if the concrete application that uses the framework adapts its functionality by different compositions of the framework's classes. In this case the concrete application uses only the built-in functionality of the framework, that is, those modules with which the framework is already equipped. In contrast, the functionality of a white-box framework is refined or extended by a concrete application, by adding additional modules through inheritance of (abstract) classes. Between these two contrasts arbitrary "shades of gray" are possible (Szyperski et al., 2002). The design of the MM4U framework lies somewhere in the middle between pure black-box and pure white-box. Being a domain independent framework, MM4U needs to be configured and extended to meet the specific requirements of a concrete personalized multimedia application. The framework provides many modules for example to access media data and associated meta data, user profile information, and generates the personalized multimedia content in a standardized output format that can be re-used for different application areas (black box usage). For the very application-specific personalization functionality, the framework can be extended correspondingly (white box usage).

The usage of the MM4U framework by a concrete personalized multimedia application is illustrated schematically in Figure 3. The personalized multimedia application uses the functionality of the framework to create personalized multimedia content, and integrates it in whatever application dependent functionality is needed, either by using the already build-in functionality of the framework or by extending it for the specific requirements of the concrete personalized multimedia application.

In respect of the multimedia software development process the MM4U framework assists the computer scientists during the design and implementation phase. It alleviates the time consuming multimedia content assembly task and lets the computer scientists concentrate on the development of the actual application. The MM4U framework provides functionality for the single tasks of the personalization engine as described in Section 3. It offers the computer scientists support for integrating and accessing user profile information and media data, selecting media elements according to the user's profile information, composing these elements into coherent multimedia content, and generating this content in standardized multimedia document formats to be presented on the user's end device.

When designing a framework, the challenge is to identify the points where the framework should be flexible, i.e., to identify the semantic aspects of the framework's application domain that have to be kept flexible. These points are the so-called *hot spots* and represent points or sockets of the intended flexibility of a framework (Pree, 1995). Each hot spot constitutes a well-defined interface where proper modules can be plugged in. When designing the MM4U framework we identified hot spots where adequate modules for supporting the personalization task can be plugged in that provide the required functionality.

As depicted in Figure 3, the MM4U framework provides four types of such hot spots, where different types of modules can be plugged in. Each hot spot represents a particular task of the personalization process. The hot spots can be realized by plugging in a module that implements the hot spot's functionality for a concrete personalized multimedia application. These modules can be both application-dependent and application-independent. For example, the access to media data and associated meta data is not necessarily application-dependent, whereas the composition of personalized multimedia content can be heavily dependent on the concrete application.

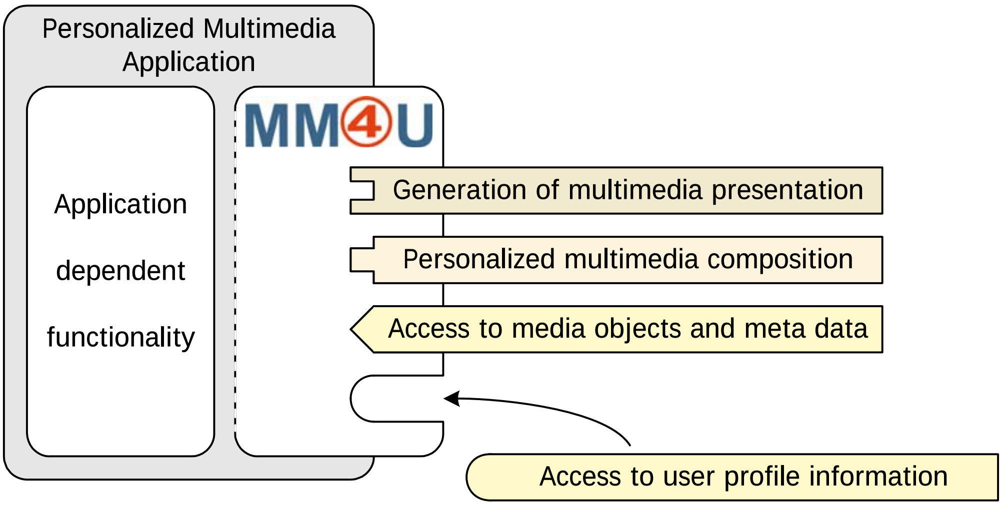

Figure 3: Usage of the MM4U framework by a personalized multimedia application

After the general design of the framework, we take a closer look at the concrete architecture of MM4U and its components in the next section.

5.3. *Design of the framework layers*

For supporting the different tasks of the multimedia personalization process, which are the access to user profile information and media data, selection and composition of media elements into a coherent presentation, rendering and display of the multimedia presentation on the end device, a layered architecture seems to be best suited for MM4U. The layered design of the framework is illustrated in Figure 4. Each layer provides modular support for the different tasks of the multimedia personalization process. The access to user profile information and media data are realized by the layers (1) and (2), followed by the two layers (3) and (4) in the middle for composition of the multimedia presentation in an internal object-oriented representation and its later transformation into a concrete presentation output format. Finally, the top layer (5) realizes the rendering and display of the multimedia presentation on the end device.

To be most flexible for the different requirements of the concrete personalized multimedia applications the framework's layer allow to extend the functionality of MM4U by embedding additional modules as indicated by the empty boxes with dots. In the following, the features of the framework are described along its different layers. We start from the bottom of the architecture and end with the top layer:

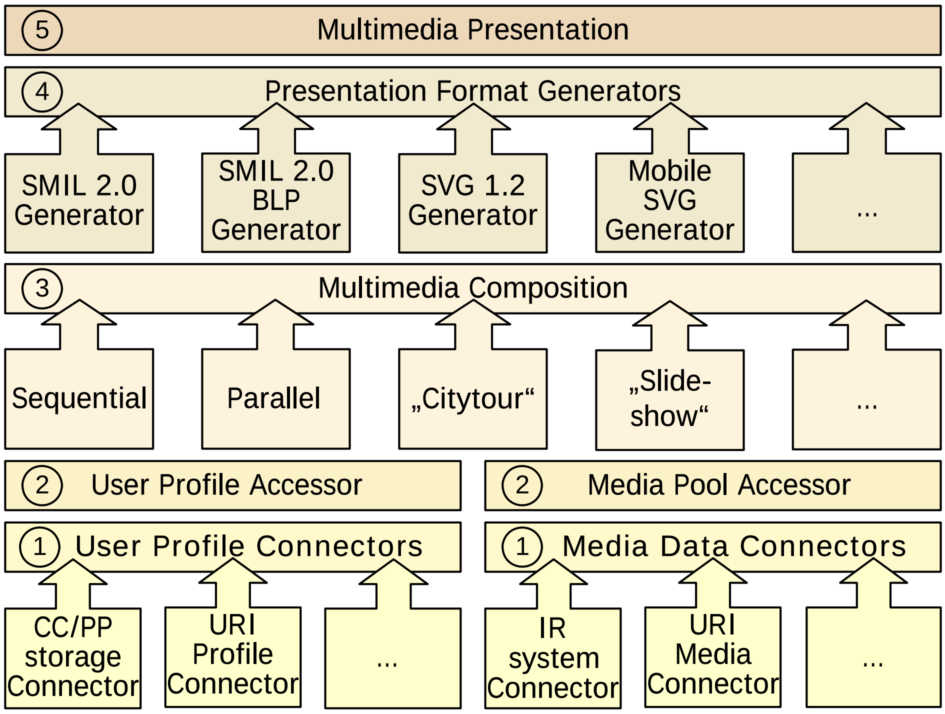

Figure 4: Overview of the multimedia personalization framework MM4U

(1) *Connectors*: The User Profile Connectors and Media Data Connectors bring the user profile data and media data into the framework. They integrate existing systems for user profile stores, media storage, and retrieval solutions. As there are many different systems and formats available for user profile information, the User Profile Connectors abstract from the actual access to and retrieval of user profile information and provide a unified interface to the profile information. With this component, the different formats and structures of user profile models can be made accessible via a unified interface. For example, a flexible URIProfileConnector we developed for our demonstrator applications gains access to user profiles over the internet. These user profiles are described as hierarchical ordered key-value pairs. This is a quite simple model but already powerful enough to allow effective pattern-matching queries on the user profiles (Chen & Kotz, 2000). However, as shown in Figure 4 also a User Profile Connector for the access to, e.g., a Composite Capability/Preference Profile (CC/PP) server could be plugged into the framework.

On the same level, the Media Data Connectors abstract from the access to media elements in different media storage and retrieval solutions that are available today with a unified interface. The different systems for storage and content-based retrieval of media data are interfaced by this component. For example the URIMediaConnector, we developed for our demonstrator applications, provides a flexible access of media objects and its associated meta data from the internet via http or ftp protocol. The meta data is stored in a single index file, describing not only the technical characteristics of the media elements and containing the location where to find the media elements in the Internet, but also comprise additional information about them, e.g., a short description of what is shown in a picture or keywords one can search for. By analogy with the access to user profile information, another Media Data Connector plugged into the framework could provide access to other media and meta data sources, e.g., an image retrieval (IR) system like IBM’s QBIC (IBM Corporation, 2004b).

The Media Data Connector supports the query of media elements by the client application (client-pull) as well as the automatic notification of the personalized application when a new media object arises in the media database (server-push). The latter is required for example by the personalized multimedia sports news ticker (see Section 8.2) which bases on a multimedia event space (Boll & Westermann, 2003).

(2) *Accessors*: The User Profile Accessor and the Media Pool Accessor provide the internal data model of the user profiles and media data information within the system. Via this layer the user profile information and media data needed for the desired content personalization is accessible and processable for the application. The Connectors and Accessors are designed such that they are *not* re-inventing existing systems for user modeling or multimedia content management. They rather provide a seamless integration of the systems by distinct interfaces and comprehensive data models.

In addition, when a personalized multimedia application uses more than one user profile database or media database the Accessor layer encapsulates the resources so that the access to them is transparent to the client application.

While the following layer (3) to (5) each constitute single components within the MM4U framework, the Accessor layer and Connectors layer do not. Instead the left side and the right side of the layers (1) and (2), i.e., the User Profile Accessor and User Profile Connectors as well as the Media Pool Accessor and Media Data Connectors, each form one component in MM4U.

(3) *Multimedia Composition*: The Multimedia Composition component comprises abstract operators in compliance with the composition capabilities of multimedia composition models like SMIL, Madeus, and ZYX, which provide complex multimedia composition functionality. It employs the data from the User Profile Accessor and the Media Pool Accessor for the multimedia composition task. The Multimedia Composition component is developed as such that it enables to develop additional, possibly more complex or application-specific composition operators that can be seamlessly plugged-in into the framework. Result of the multimedia composition is an internal object-oriented representation of the personalized multimedia content independent of the different presentation formats.

(4) *Presentation Format Generators*: The Presentation Format Generators work on the internal object-oriented data model provided by the Multimedia Composition component and convert it into a standardized presentation format that can be displayed by the corresponding multimedia player on the client device. In contrast to the multimedia composition operators, the Presentation Format Generators are completely independent of the concrete application domain and only rely on the targeted output format. In MM4U, we already developed Presentation Format Generators for SMIL 2.0, the Basic Language Profile (BLP) of SMIL 2.0 for mobile devices (Ayars et al., 2001), SVG 1.2, Mobile SVG 1.2 (Andersson et al., 2004a) comprising SVG Tiny for multimedia-ready mobile phones and SVG Basic for pocket computers like Personal Digital Assistants (PDA) and Handheld Computers (HHC), and HTML (Raggett et al., 1998). We are currently working on Presentation Format Generators for Macromedia Flash (Macromedia, 2004) and other multimedia document model formats like for example HTML+TIME, the 3GPP SMIL Language Profile (3rd Generation Partnership Project, 2003b), which is a subset of SMIL used for scene description within the Multimedia Messaging Service (MMS) interchange format (3rd Generation Partnership Project, 2003a), and XMT-Omega, a high level abstraction of MPEG-4 based on SMIL (Kim et al., 2000).

(5) *Multimedia Presentation*: The Multimedia Presentation component on top of the framework realizes the interface for applications to actually play the presentation of different multimedia presentation formats. The goal here is to integrate existing presentation components of the common multimedia presentation formats like SMIL, SVG, or HTML+TIME which the underlying Presentation Format Generator produces. So the developers benefit from the fact that only players for standardized multimedia formats need to

be installed on the user's end device and that they must not effort any time and resources in developing an own render and display engine for their personalized multimedia application.

The layered architecture of MM4U allows being easily adapted for the particular requirements that can occur in the development of personalized multimedia applications. So special user profile connectors as well as media database connectors can be embedded into the Connectors layer of the MM4U framework to integrate the most diverse and individual solutions for storage, retrieval and, gathering for user profile information and media data. With the ability to extend the Multimedia Composition layer by complex and sophisticated composition operators, arbitrary personalization functionality can be added to the framework. The Presentation Format Generator component allows integrating any output format into the framework to support the most different multimedia players that are available for the different end devices.

The personalized selection and composition of media elements and operators into a coherent multimedia presentation is the central task of the multimedia content creation process which we present in more detail in the following section.

6. CREATING PERSONALIZED MULTIMEDIA CONTENT

The MM4U framework provides the general functionality for the dynamic composition of media elements and composition operators into a coherent personalized multimedia presentation. Having presented the framework layers in the previous section, we now look in more detail how the layers contribute to the different tasks in the general personalization process as shown in Figure 2. The Media Data Accessor layer provides the personalized selection of media elements by their associated meta data and is described in Section 6.1. The Multimedia Composition layer supports the composition of media elements into time and space in the internal multimedia representation format in three different manners, which are presented in detail in the Sections 6.2 to 6.4. Finally, Section 6.5 describes the last step, the transformation of the multimedia content in internal document model to an output format that is actually delivered to and rendered by the client devices. This is supported by the Presentation Format Generators layer.

6.1. Personalized multimedia content selection

For creating personalized multimedia content first those media elements have to be selected from the media databases that are most relevant to the user's request. This personalized media selection is realized by the Media Data Accessor and Media Data Connector component of the framework. For the actual personalized selection of media elements a context object is created within the Multimedia Composition layer carrying the user profile information, technical characteristics of the end device, and further application-specific information. With this context object, the unified interface of the Media Data Accessor for querying media elements is called. The context object is handed over to the concrete Media Data Connector of the connected media database. Within the Media Data Connector, the context object is mapped to the meta data associated with the media elements in the database and those media elements are determined that match the request, i.e., the given context object at best. It is important to note that the Media Data Accessor and Media Data Connector layer integrate and embrace existing multimedia information systems and modern content based multimedia retrieval solutions. This means that the retrieval of the "best match" can only be left to the underlying storage and management systems. The framework can only provide for comprehensive and lean interfaces to these systems. This can be our own URIMediaServer accessed by the URIMediaConnector but also other multimedia databases or (multi)media retrieval solutions. The result set of the query is handed back by the Accessor to the composition layer.

For example, the context object for our mobile tourist guide application carries information about user interests and preferences in respect to the sights of the city, the display size of the end device, and the location for the tourist guide. The Media Data Connector, realized in this case by our URIMediaConnector, processes this context object and returns images and videos from those sights in Oldenburg that both match the user's interests and preferences as well as the limited display size of the mobile device.

Based on the personalized selection of media elements the Multimedia Composition layer provides the assembly of these media elements on three different manners, the basic, complex and sophisticated composition of multimedia content, which is described in the following sections.

6.2. Basic composition functionality

With the basic composition functionality the MM4U framework provides the basic bricks for composing multimedia content. It forms the basis for assembling the selected media elements into personalized multimedia documents and provides the means for realizing the central aspects of multimedia document models, i.e., the temporal model, the spatial layout, and the interaction possibilities of the multimedia presentation. The temporal model of the multimedia presentation is determined by the temporal relationships between the presentation's media elements formed by the composition operators. The spatial layout expresses the arrangement and style of the visual media elements in the multimedia presentation. Finally, with the interaction model the user interaction of the multimedia presentation is determined, in order to let the user choose between different paths of a presentation. For the temporal model, we selected an interval-based approach as it can be found in (Duda & Keramane, 1995). The spatial layout is realized by a hierarchical model for media positioning (Boll & Klas, 2001). For interaction with the user navigational and decision interaction are supported, as can be found with SMIL (Ayars et al., 2001) and MHEG-5 (Echiffre et al., 1998; International Organisation for Standardization, 1996).

A basic composition operator or short a *basic operator* can be regarded as an atomic unit for multimedia composition, which can not be further broken down. Basic operators are quite simple but applicable for any application area and therefore most flexible. Basic temporal operators realize the temporal model and basic interaction operators realize the interaction possibilities of the multimedia presentation, as specified above. The two basic temporal operators *Sequential* and *Parallel*, e.g., can be used to present media elements one after the other in a sequence respectively to present media elements parallel at the same time. With basic temporal operators and media elements the temporal course of the presentation can be determined like for example a slideshow as depicted in Figure 5. The operators are represented by white rectangles and the media elements by gray ones. The relation between the media elements and the basic operators is shown by the edges beginning with a filled circle at an operator and ending with a filled rhombus respectively a diamond at a media element or another operator. The semantics of the slideshow shown in Figure 5 is that it starts with the presentation of the root element, which is the Parallel operator. The semantics of the Parallel operator is that it shows the operators and media elements that are attached to it at the same time. This means that the audio file starts to play while simultaneously the Sequential operator is presented. The semantics of the Sequential operator is to show the attached media elements one after another, so while the audio file is played in background the four slides are presented in sequence.

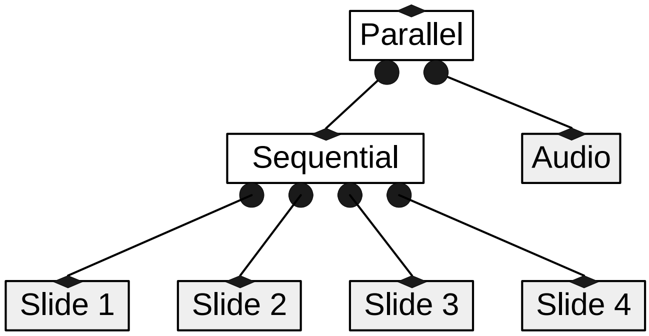

Figure 5: Slideshow as an example of assembled multimedia content

Besides the basic composition *operators* the so-called *projectors* are part of the Multimedia Composition layer. Projectors can be attached to operators and media elements to define, e.g., the visual and acoustical layout of the multimedia presentation. Figure 6 shows the slideshow example from above with projectors attached. The spatial position as well as the width and height of the single slide media elements are determined by the corresponding *SpatialProjectors*. The volume, treble, bass, and balance of the audio medium is determined by the attached *AcousticProjector*.

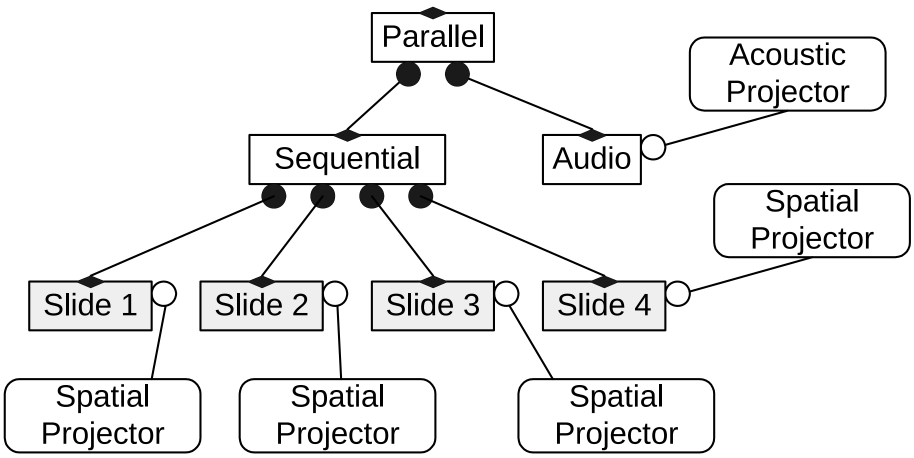

Figure 6: Adding layout to the slideshow example

Besides temporal operators the Multimedia Composition component offers basic operators for specifying the interaction possibilities of the multimedia presentation. Interaction can be added, e.g., by using the basic operator *InteractiveLink*. It defines a link represented by a single media element or a fragment of a multimedia presentation that is clickable by the user and a target presentation the user receives if he or she clicks on the link.

The description above presents some examples of the basic composition functionality the MM4U framework offers. The framework comprises customary composition operators for creating multimedia content as provided by modern multimedia presentation formats like SMIL and SVG. Even though the basic composition functionality does not reflect the fancy features of some of today's multimedia presentation formats, it supports the very central multimedia features of modeling time, space, and interaction. This allows the transformation of the internal document model into many different multimedia presentation formats for different end devices.

With the basic multimedia composition operators the framework offers, arbitrary multimedia presentations can be assembled. However, so far the MM4U framework provides "just" basic multimedia composition functionality. In the same way as one would use an authoring tool to create SMIL presentations, for example the GR*i*NS editor (Oratrix, 2004), one can also use a corresponding authoring tool for the basic composition operators the MM4U framework offers to create multimedia content. For reasons of reusing parts of the created multimedia presentations, e.g., a menu bar or a presentation's layout, and for convenience, there is a need for more complex and application-specific composition operators that provide a more convenient support for creating the multimedia content.

6.3. Complex composition functionality

For creating presentations that are more complex the Multimedia Composition layer provides the ability to abstract from basic to complex operators. A complex composition operator encapsulates the composition functionality of an arbitrary number of basic operators and projectors and provides the developers with a more complex and application-specific building block for creating the multimedia content. Complex composition operators are composed of basic and others complex operators. As complex composition operators not only embed basic but also other complex operators they provide for a reuse of composition operators. In contrast to the basic operators, the complex composition operators can be dismantled into its individual parts. Figure 7 depicts a complex composition operator for our slideshow example above. It encapsulates the media elements, operators, and projectors of the slideshow (the latter are omitted in the diagram to reduce complexity). The complex operator *Slideshow*, indicated by a small "c"-symbol in the upper right corner, represents an encapsulation of the former slideshow presentation in a complex object and forms itself a building block for more complex multimedia composition.

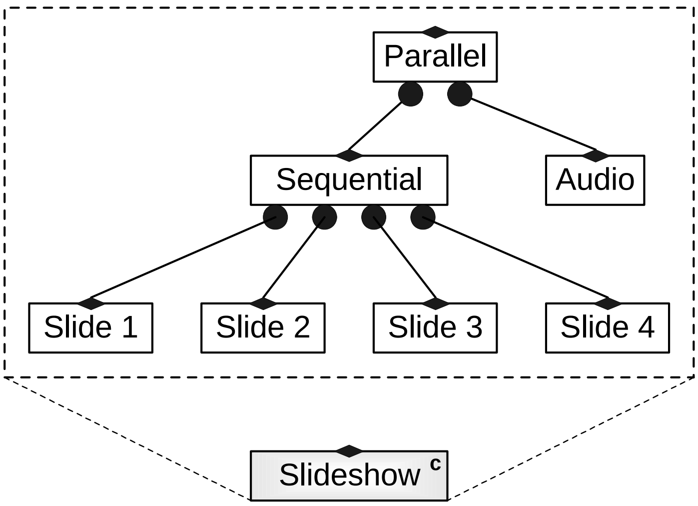

Figure 7: The slideshow example as a complex composition operator

Complex operators as described above, define fixed encapsulated presentations. Their temporal flow, spatial layout, and the used media elements cannot be changed subsequently. However, a complex composition operator does not necessarily need to specify all media elements, operators, and projectors of the respective multimedia document tree. Instead, to be more flexible, some parts can be intentionally left open. These parts constitute the parameters of a complex composition operator and have to be filled in for concrete usage of these operators. Such parameterized complex composition operators are one means to define *multimedia composition templates* within the MM4U framework. However, only pre-structured multimedia content can be created with these templates, since the complex composition operators can only encapsulate presentations of a fixed structure.

Figure 8 shows the slideshow example as a parameterized complex composition operator. In this case, the complex operator Slideshow comprises the two basic operators Parallel and Sequential. The Slideshow's parameters are the place holders for the single slides and have to be instantiated when the operator is used within the multimedia composition. The slideshow's audio file is already pre-selected. In addition, the parameters of a complex composition operator can be typed, i.e., they expect a special type of operator or media element. The Slideshow operator would expect visual media elements for the parameters *Slide 1* to *Slide 4*. To indicate the complex operator's parameters, they are visualized by rectangles with dotted lines. The pre-selected audio file is already encapsulated in the complex operator as illustrated in Figure 7.

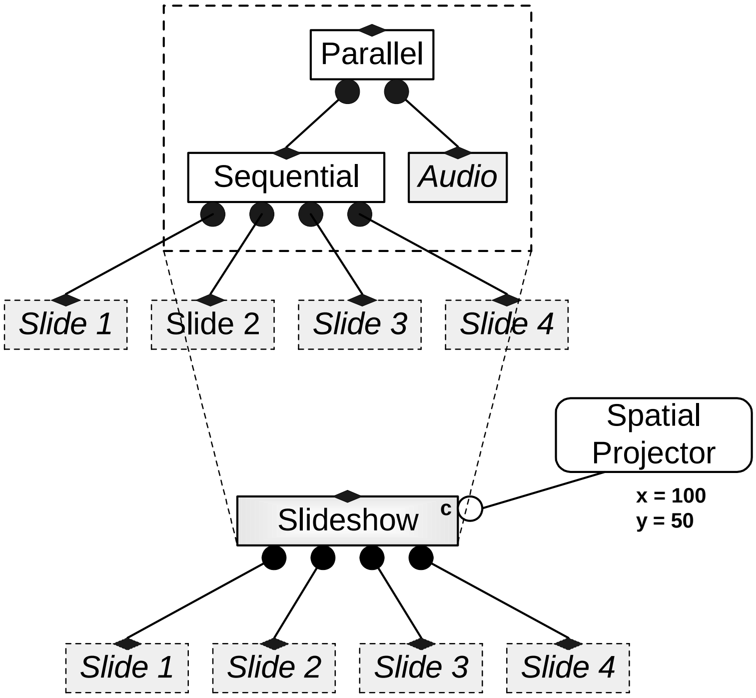

Figure 8: The slideshow example as parameterized complex composition operator

In the same way as projectors are attached to basic operators in Section 6.2, they can also be attached to complex operators. The SpatialProjector attached to the Slideshow operator shown in Figure 8 determines that the slideshow's position within a multimedia presentation is the position x = 100 pixel and y = 50 pixel in relation to the position of its parent node.

With basic and complex composition operators one can build multimedia composition functionality that is equivalent to the composition functionality of advanced multimedia document models like Madeus (Jourdan, 1998) and ZYX (Boll & Klas, 2001). Though complex composition operators can have an arbitrary number of parameters and can be configured individually each time they are used, the internal structure of complex operators is still static. Once a complex operator is defined, the number of parameters and their type are fixed and can not be changed. Using a complex composition operator can be regarded as filling in fixed composition templates with suitable media elements. Personalization can only take place in selecting those media elements that fit the user profile information at best. For the dynamic creation of *personalized* multimedia content even more sophisticated composition functionality is needed, that allows the composition operators to change the structure of the generated multimedia content at runtime. To realize such sophisticated composition functionality additional composition logic needs to be included into the composition operators, which can not expressed anymore even by the mentioned advanced document models we find in the field.

6.4. *Sophisticated composition functionality*

With basic and complex composition functionality, we already provide the dynamic composition of pre-structured multimedia content by parameterized multimedia composition templates. However, such templates are only flexible concerning the selection of the concrete composition parameters. To achieve an even more flexible dynamic content composition, the framework provides sophisticated composition operators, which allow determining the document structure and layout during creation time by additional composition logic. Multimedia composition templates defined by using such sophisticated composition operators are no longer limited to create pre-structured multimedia content only, but determine the document structure and layout of the multimedia content on-the-fly and depending on the user profile information, the characteristics of the used end device, and any other additional information. The latter can be, e.g., a database containing sightseeing information. Such sophisticated composition operators exploit the basic and complex composition operators the MM4U framework offers but allow more flexible, possibly application-specific multimedia composition and personalization functionality with their additional composition logic. This composition logic can be realized by using document structures, templates, constraints and rules, or by plain programming. Independent of how the sophisticated multimedia content composition functionality is actually realized, the result of this composition process is always a multimedia document tree that consists of basic and complex operators, projectors, as well as media elements. In our graphical notation, sophisticated composition operators are represented in the same way as complex operators, but are labeled with a small "s" symbol in the upper right corner.

Figure 9 shows an example of a parameterized sophisticated composition operator, the *CityMap*. This operator provides the generation of a multimedia presentation containing a city map image together with a set of available sightseeing spots on it. The parameters of this sophisticated operator are the city map image of arbitrary size and a spot image used for presenting the sights on the map. Furthermore, the CityMap operator reads out the positions of the sights on a reference map (indicated by the table on the right) and automatically re-calculates the positions in dependence of the size of the actual city map image. Which spots are selected and actually presented on the city map depends on the user profile, in particular the types of sights he or she is interested in, and the categories a sight belongs to. In addition, the size of the city map image is selected to fit the display of the end device at best. The CityMap operator is used within our personalized city guide prototype presented in Section 8.1 and serves there for Desktop PCs as well as mobile devices.

The multimedia document tree generated by the CityMap operator is shown in the bottom part of Figure 9. Its root element constitutes the Parallel operator. Attached to it are the image of the city map and a set of InteractiveLink operators. Each InteractiveLink represents a spot on the city map, instantiated by the spot image. The user can click on the spots to receive multimedia presentations with further information about the sights. The positions of the spot images on the city map are determined by the SpatialProjectors. The personalized multimedia presentations about the sights are represented by the sophisticated operators Target 1 to Target N.

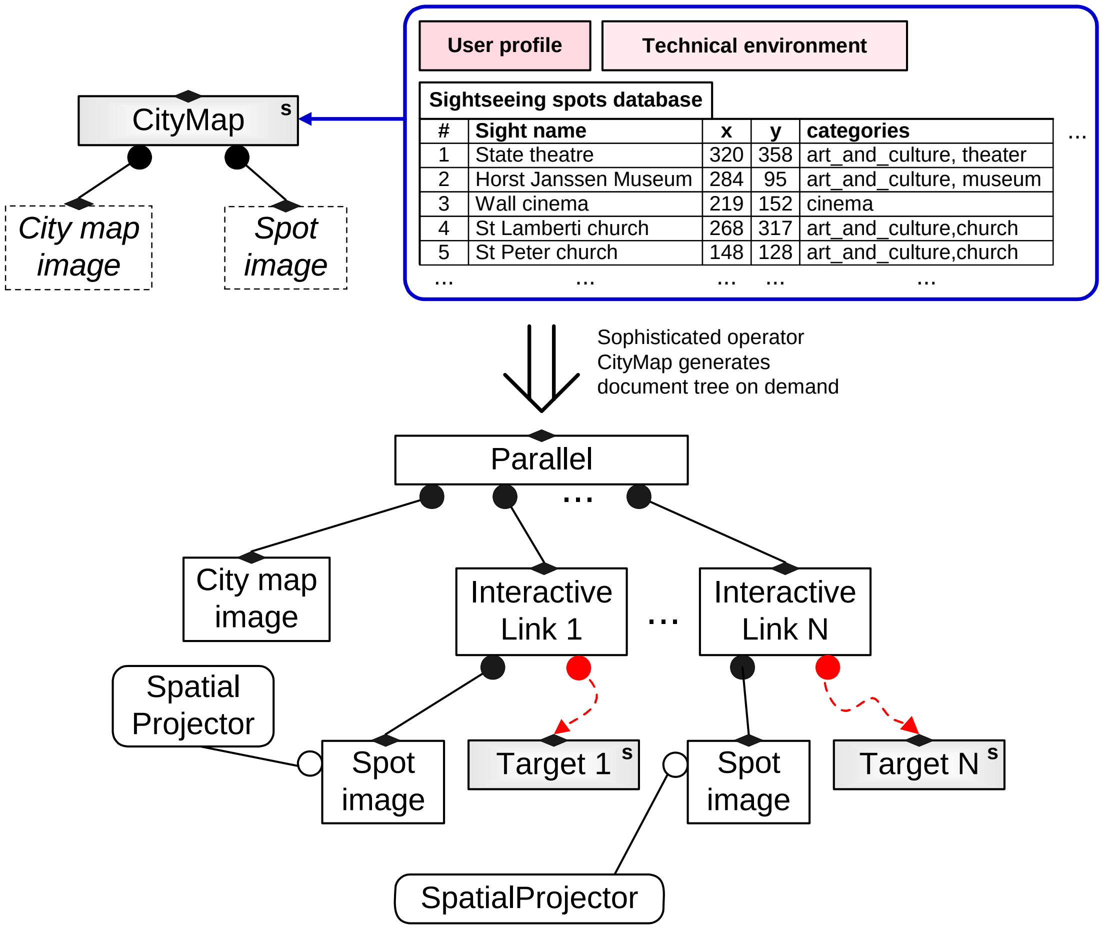

Figure 9: Insight into the sophisticated composition operator CityMap

The CityMap operator is one example of extending the personalization functionality of the MM4U framework by a sophisticated application-specific multimedia composition operator, here in the area of (mobile) tourism applications. This operator, e.g., is developed by programming the required dynamic multimedia composition functionality. However, the realization of the internal composition logic of sophisticated operators is independent of the used technology and programming language. The same composition logic could also be realized by using a different technology, as for example a constraint-based approach. Though the actual realization of the personalized multimedia composition functionality would be different, the multimedia document tree generated by this rule based sophisticated operator would be the same as depicted in Figure 9.

Sophisticated composition operators allow embracing the most different solutions for realizing personalized multimedia composition functionality. This can be plain programming as in the example of the CityMap operator, document structures and templates that are dynamically selected according to the user profile information and filled in with media elements relevant to the user, or systems describing the multimedia content generation by constraints and rules.

The core MM4U framework might not offer all kinds of personalized multimedia composition functionality one might require, since the personalization functionality always depends on the actual application to be developed and thus can be very specific. Instead the framework provides the basis to develop sophisticated personalized multimedia composition operators, such that every application can integrate its own personalization functionality into the framework. So, every sophisticated composition operator can be seen as a "small application" itself that can conduct a particular multimedia personalization functionality. This "small

application" can be re-used within others and thus extends the functionality of the framework. The Multimedia Composition component allows a seamless plug-in of arbitrary sophisticated composition operators into the MM4U framework. This enables the most complex personalized multimedia composition task to be "just" plugged into the system and to be used by a concrete personalized multimedia application.
With the sophisticated composition operators the MM4U framework provides its most powerful and flexible functionality to generate arbitrary personalized multimedia content. However, this multimedia content is still represented in an internal document model and has to be transformed into a presentation format that can be rendered and displayed by multimedia players on the end devices.

6.5. From multimedia composition to presentation

In the last step of the personalization process, the personalized multimedia content represented in our internal document model is transformed by the Presentation Format Generators component into one of the supported standardized multimedia presentation formats, which can be rendered and displayed on the client device. The output format of the multimedia presentation is selected according to the user's preferences and the capabilities of the end device, i.e., the available multimedia players and the multimedia presentation formats they support.

The Presentation Format Generators adapt the characteristics and facilities of the internal document model provided by the Multimedia Composition layer in regard of the used time model, spatial layout, and interaction possibilities to the particular characteristics and syntax of the concrete presentation format. For example, the spatial layout of our internal document model is realized by a hierarchical model that supports the positioning of media elements in relation to other media elements. This relative positioning is supported by most of today's presentation formats, e.g., SMIL, SVG, and XMT-Ω. However, there exist multimedia presentation formats that do not support such a hierarchical model and only allow an absolute positioning of visual media elements in regard of the presentation's origin, as for example, the Basic Language Profile of SMIL, 3GPP SMIL, and Macromedia's Flash. In this case, the Presentation Format Generators component transforms the hierarchically organized spatial layout of the internal document model to a spatial layout of absolute positioning. How the transformation of the spatial model is actually performed and how the temporal model and interaction possibilities of the internal document model are transformed into the characteristics and syntax of the concrete presentation formats is intentionally omitted in this book chapter due to its focus on the composition and assembly of the personalized multimedia content and is described in (Scherp & Boll, 2005).

7. IMPACT OF PERSONALIZATION TO THE DEVELOPMENT OF MULTIMEDIA APPLICATIONS

The multimedia personalization framework MM4U presented so far provides support to develop sophisticated personalized multimedia applications. Involved parties in the development of such applications is typically a heterogeneous team of developers from different fields including media designers, computer scientists, and domain experts. In this section, we describe what challenges *personalization* brings to the development of personalized multimedia applications and how and where the MM4U framework can support the developer team to accomplish their job.

In Section 7.1 the general software engineering issues in regard of personalization are discussed. We describe how personalization affects the single members of the heterogeneous developer team and how the MM4U framework supports the development of personalized multimedia applications. The challenges that arise with creating personalized multimedia content by the domain experts using an authoring tool are presented in Section 7.2. We also introduce how the MM4U framework can be used to develop a domain-specific authoring tool in the field of e-learning content, which aims to hide the technical details of content authoring from the authors and lets them concentrate on the actual creation of the personalized multimedia content.

7.1. Influence of personalization to multimedia software engineering

We observe that software engineering support for multimedia applications such as proper process models and development methodologies are sparsely to be found in this area. Furthermore, the existing process models and development methodologies for multimedia applications as for example (Rout & Sherwood, 1999) and (Engels et al., 2003) do not support personalization aspects. However, personalization requirements complicate the software development process even more and increase the development costs, since every individual alternatives and variants have to be anticipated, considered, and actually implemented. Therefore, there is a high demand in supporting the development process of such applications. In the following, we first introduce how personalization affects the software development process respectively the multimedia content creation process in general. Then we identify what support the developers of personalized multimedia applications need and consider where the MM4U framework supports the development process.

Since the term *personalization* profoundly depends on the application's context, its meaning has ever to be reconsidered when developing a personalized application for a new domain. (Rossi et al., 2001) claim, that personalization should be considered directly from the beginning when a project is conceived. Therefore, the first activity when developing a personalized multimedia application is to determine the personalization requirements, i.e., which aspects of personalization should be supported by the actual application. For example, in the case of an e-learning application the personalization aspects consider the automatic adaptation to the different learning styles of the students and their prior knowledge about the topic. In addition, different degrees of difficulty should be supported by a personalized e-learning application. In the case of a personalized mobile tourism application, however, the user's location and his or her surrounding would be of interest for personalization instead. These personalization aspects must be kept in mind during every activity throughout the whole development process. The decision which personalization aspects are to be supported has to be incorporated in the analysis and design of the personalized application and will hopefully entail to a flexible and extendible software design. However, this increases the overall complexity of the application to be developed and automatically leads to a higher development effort including longer development duration and higher costs. Therefore, a good requirement analysis is crucial when developing personalized applications lest dissipate one's energies in bad software design in respect of the personalization aspects.

When transferring the requirements for developing personalized software to the specific requirements of personalized multimedia applications one can say that it affects all members of the developer team: the domain expert, the media designers, and the computer scientists, and putting higher requirements to them.

The domain expert normally contributes to the development of multimedia applications by providing input to draw storyboards of the specific application's domain. These storyboards are normally drawn by media designers and are the most important means to communicate the later application's functionality within the developer team. When personalization comes into account, it is difficult to draw such storyboards, because of the many possible alternatives and different paths in the application that are implicated with personalization. Consequently, the storyboards change in regard of, e.g., the individual user profiles and the end devices that are used. When drawing storyboards for a *personalized* multimedia application, those points in the storyboard have to be identified and visualized where personalization is required and needed. Storyboards have to be drawn for every typical personalization scenario concerning the concrete application. This drawing task should be supported by interactive graphical tools to create "personalized" storyboards and to identify reusable parts and modules of the content.

It is the task of the media designer in the development of multimedia applications to plan, acquire, and create media elements. With personalization, media designers have to think additionally about the usage of media elements for personalization purposes, i.e., the media elements have to be created and prepared for different contexts. When acquiring media elements, the media designers must consider for which user context the media elements are created and what aspects of personalization are to be supported, e.g., different styles, colours, and spatial dimensions. Possibly a set of quite similar media assets have to be developed, that only differ in certain aspects. For example, an image or video has to be transformed for different end device resolutions, colour depth, and network connections. Since personalization means to (re)assemble existing media elements into a new multimedia presentation, the media designers will also have to identify reusable media elements. This means that additionally the storyboards must already capture the personalization aspects. Not only the content but also the layout of the multimedia application can change in dependence of the user context. So, the media designers have to create different visual layouts for the same application to serve the needs of different user groups. For example an e-learning system for children would generate colourful multimedia presentations with many auditory elements and a comic-like virtual assistant, whereas the system would present the same content much more factual for adults. This short discussion shows that personalization already affects the storyboarding and media acquisition. Creating media elements for personalized multimedia applications requires a better and elaborate planning of the multimedia production. Therefore, a good media production strategy is crucial, due to the high costs involved with the media production process. Consequently, the domain experts and the media designers need to be supported by appropriate tools for planning, acquiring, and editing media elements for personalized multimedia applications.

The computer scientists actually have to develop the multimedia personalization functionality of the concrete application. What this personalization functionality is depends heavily on the concrete application domain and is communicated with the domain experts and media designers by using personalized storyboards. With personalization, the design of the application has to be more flexible and more abstract to meet the requirements of changing user profile information and different end device characteristics. This is where the MM4U framework comes into play. It provides the computer scientists the general architecture of the personalized multimedia application and supports them in designing and implementing the concrete multimedia personalization functionality. When using the MM4U framework, the computer scientists must know how to use and to extend it. The framework provides the basis for developing both basic and sophisticated multimedia personalization functionality, as for example the Slideshow or the CityMap operator presented in Section 6. To assist the computer scientists methodically we currently work on guidelines and checklists of how to develop the personalized multimedia composition operators and how to apply them. Consequently, the development of personalized multimedia applications by using the MM4U framework basically means to the computer scientists the design, development, and deployment of multimedia composition operators for generating personalized content. The concept of the multimedia personalization operators as introduced in Section 6.4, that every concrete personalized multimedia application is itself a new composition operator increases re-usage of existing personalization functionality. Furthermore, the interface design of the sophisticated operators makes it possible to embrace existing approaches that are able to generate multimedia document trees, e.g., like it can be generated with the basic and complex composition functionality of the MM4U framework.

7.2. Influence of personalization to multimedia content authoring

Authoring of multimedia content is the process in which the multimedia presentations are actually created. This creation process is typically supported by graphical authoring tools, e.g., Macromedia's Authorware and Director (Macromedia, 2004), Toolbook (Click2learn, 2001-2002), (Arndt, 1999), and (Gaggi & Celentano, 2002). For creating the multimedia content, the authoring tools follow different design philosophies and metaphors, respectively. These metaphors can be roughly categorized into script-based, card/page-based, icon-based, timeline-based, and object-based authoring (Rabin & Burns, 1996). All these different metaphors have the same goal, to support authors in creating their content. Even though based on these metaphors a set of valuable authoring tools has been developed, these metaphors do not necessarily provide a suitable means for authoring personalized content.

From the context of our research project Cardio-OP we derived early experiences with personalized content authoring for domain experts in the field of cardiac surgery (Klas et al., 1999; Greiner & Rose, 1998; Boll et al., 2001). One of the tools developed by a project partner, the Cardio-OP Authoring Wizard, is a page-based easy-to-use multimedia authoring environment, enabling medical experts to compose a multimedia book on operative techniques in the domain of cardiac surgery for three different target groups, medical doctors, nurses, and students. The Authoring Wizard guides the author through the particular authoring steps and offers dialogues specifically tailored to the needs of each step. Coupled tightly with an underlying media server, the authoring wizard allows to employ every precious piece of media data available at the media server in every of the instructional applications at different educational levels. This promotes reuse of expensively produced content in a variety of different contexts.

Personalization of the e-learning content is required here, since the three target groups have different views and knowledge about the domain of cardiac surgery. Therefore, the target groups require different information from such a multimedia book, presented on an adequate level of difficulty for each group.

However, the experiences we gained from deploying this tool show that it is hard to provide the domain authors with an adequate intuitive user interface for the creation of personalized multimedia e-learning content for three educational levels. It was a specific challenge for the computer scientists involved in the project to provide both media creation tools and multimedia authoring wizard that allow the domain experts to insert knowledge into the system, while at the same time hiding as much of the technical details from then.

On the basis of the MM4U framework, we are currently developing a "smart authoring tool" aimed for domain experts to create personalized multimedia e-learning content. The tool we develop works at the what-you-see-is-what-you-get (WYSIWYG) level and can be seen as a specialized application employing the framework to create personalized content. The content source from which these personalized e-learning content is created, constitute the LEBONED repositories. Within the LEBONED project (Oldenettel & Malachinski, 2003) digital libraries are integrated into learning management systems. Using the content managed by the LEBONED system for new e-learning units, a multimedia authoring support is needed for assembling existing e-learning modules into new, possibly more complex units. In the e-learning context, the background of the learners is very relevant for the content that meets the users learning demands – that means a personalized multimedia content can meet the user's background knowledge and interest much better than a one-size-fits-all e-learning unit. The creation of an e-learning unit on the other side can not be supported by a mere automatic process. Rather the domain experts would like to control the assembly of the content as they are responsible for the content conveyed. The smart authoring tool guides the domain experts through the composition process and supports them in creating presentations that still provide flexibility to the targeted user context. In the e-learning context we can expect domain experts such as lecturers that want to create a new e-learning unit but do not want to be bothered with the technical details of (multimedia) authoring.

We use the MM4U framework to build the multimedia composition and personalization functionality of this smart authoring tool. For this, the Multimedia Composition component supports the creation and processing of arbitrary document structures and templates. The authoring tool exploits this functionality for composition to achieve a document structure that is suitable just for that content domain and the targeted audience. The Media Data Accessor supports the authoring tool in those parts in which it lets the author to choose from only those media elements that are suitable for the intended user contexts and that can be adapted to the user's infrastructure. Using the Presentation Format Generators the authoring tool finally generates the presentations for the different end devices of the targeted users. Thus the authoring process is guided and specialized with regard to selecting and composing personalized multimedia content. For the development of this authoring tool, the framework fulfils the same function in the process of creating personalized multimedia content in a multimedia application as described in Section 5. However, the creation of personalized content is not achieved at once but step by step during the authoring process.

8. IMPLEMENTATION AND PROTOTYPICAL APPLICATIONS

The framework, its components, classes, and interfaces, are specified using the Unified Modeling Language and has been implemented in Java. The development process for the framework is carried out as an iterative software development with stepwise refinement and enhancement of the framework's components. The re-design phases are triggered by the actual experience of implementing the framework but also by employing the framework in several application scenarios. In addition, we are planning to provide a beta version of the MM4U framework to other developers for testing the framework and to develop their own personalized multimedia applications with MM4U.

Currently, we are implementing different application scenarios to prove the applicability of MM4U in different application domains. These prototypes are the first stress test for the framework. At the same time the development of the sample applications gives us an important feedback about the comprehensiveness and the applicability of the framework. In the following sections, two of our prototypes that base on the MM4U framework are introduced: In Section 8.1 a prototype of a personalized city guide is presented and in Section 8.2 a prototype of a personalized multimedia sports news ticker is described.

8.1. Sightseeing4U – a generic personalized city guide

Our first prototype using the MM4U framework is Sightseeing4U, a generic personalized city guide application (Scherp & Boll, 2004a+b; Boll et al., 2004). It is applicable to develop personalized tourist guides for arbitrary cities, both for Desktop PCs and mobile devices. The generic Sightseeing4U application uses the MM4U framework and its modules as depicted in Figure 3. The concrete demonstrator we developed for our home town Oldenburg in Northern Germany considers the pedestrian zone and comprises video and image material of about 50 sights. The demonstrator is developed for Desktop PCs as well as PDAs (Scherp & Boll, 2004a). It supports personalization in respect of the user's interests, e.g., churches, museums, and theatres, and preferences such as the favorite language. Depending on the specific sightseeing interests the proper sights are automatically selected for the user. This is realized by category matching of the user's interests with the meta data associated to the sights. Figure 10 and Figure 11 show some screenshots of our city guide application in different output formats and on different end devices. The presentation in Figure 10 is targeted at a user interested in culture, whereas the presentation in Figure 11 is generated for a user who is hungry and searches for a good restaurant in Oldenburg. The different interests of the users result in different spots that are presented on the map of Oldenburg. When clicking on a certain spot the user receives a multimedia presentation with further information about the sight (see the little boxes where the arrows point at). Thereby, the media elements for the multimedia presentation are automatically selected to fit the end devices characteristics best. For example, a user sitting at a Desktop PC receives a high-quality video about the palace of Oldenburg as depicted in Figure 10a, while a mobile user gets a smaller video of less quality in Figure 10b. In the same way, the user searching for a good restaurant in Oldenburg receives either a high-quality video when using a Tablet PC as depicted in Figure 11a, or a smaller one that meets the limitations of the mobile device as shown in Figure 11b. If there is no video of a particular sight available at all, the personalized tourist guide automatically selects images instead and generates a slideshow for the user.

a) RealOne Player (RealNetworks, 2003) on a Desktop PC

b) PocketSMIL Player (INRIA, 2003) on a PDA

Figure 10: Screenshots of the city guide application for a user interested in culture (presentation generated in SMIL 2.0 and SMIL 2.0 BLP format, respectively)

a) Adobe SVG Plug-In (Adobe Systems, 2001) on a Tablet PC

b) Pocket eSVG viewer (EXOR, 2001-2004) on a PDA

Figure 11: Screenshots of the Sightseeing4U prototype for a user searching for a good restaurant (output generated in SVG 1.2 and Mobile SVG format, respectively)

8.2. Sports4U – a personalized multimedia sports news ticker

A second prototype that uses our MM4U framework is the personalized multimedia sports news ticker called Sports4U. The Sports4U application exploits the MediÆther multimedia event space as introduced in (Boll & Westermann, 2003). The MediÆther bases on a decentralized peer-to-peer infrastructure and allows to publish, to find, and to be notified about any kind of multimedia events of interest. In the case of Sports4U, the event space forms the media data basis of sports related multimedia news events. A multimedia sports news event comprises data of different media types like a describing text, a title, one or more images, an audio record, or a video clip. The personalized sports ticker application combines the multimedia data of the selected events, the available meta data, and additional information, e.g., from a soccer player database. The application uses a sophisticated composition operator that automatically arranges these multimedia sports news to a coherent presentation. It regards possible constraints like running time limit and particular characteristics of the end device, like the limited display size of a mobile device. The result is a sports news presentation that can be, for example, viewed with a SMIL player over the Web as shown in Figure 12. With a suitable Media Data Connector the MediÆther is connected to the MM4U framework. This connector not only allows querying for media elements like the URIMediaConnector but also provides the notification of incoming multimedia events to the actual personalized application. Depending on the user context, the Sports4U prototype receives those sports news from the pool of sports events in the MediÆther that match the users profile. The Sports4U application alleviates the user from the time consuming task of searching for sports news he or she might be interested in.

Figure 12: Screenshots of the personalized sports newsticker Sport4U

9. CONCLUSION

In this chapter, we presented an approach for supporting the creation of personalized multimedia content. We motivated the need of technology to handle the flood of multimedia information that allows for a much targeted, individual management and access to multimedia content. To give a better understanding of the content creation process we introduced the general approaches in multimedia data modeling and multimedia authoring as we find it today. We presented how the need of personalization of multimedia content, heavily affects the multimedia content creation process and can only result in a dynamic, (semi) automatic support for the personalized assembly of multimedia content. We looked into existing related approaches ranging from personalization in the text-centric Web context over single media personalization to the personalization of multimedia content. Especially for complex personalization tasks we observe that an (additional) programming is needed and propose a software engineering support with our Multimedia for you Framework (MM4U).

We presented the MM4U framework concept in general and, in more detail, the single layers of the MM4U framework: access to user profile information, personalized media selection by meta data, composition of complex multimedia presentations, and generation of different output formats for different end devices. As central part of the framework, we developed multimedia composition operators which create multimedia content in an internal model and representation for multimedia presentations, integrating the composition capabilities of advanced multimedia composition models. Based on this representation, the framework provides so-called generators to dynamically create different context-aware multimedia presentations in formats such as SMIL and SVG. The usage of the framework and its advantages has been presented in the context of multimedia application developers but also in the specific case of using the framework's specific features for the development of a high level authoring tool for domain experts.

With the framework developed, we achieved our goals concerning the development of a domain independent framework that supports the creation of personalized multimedia content independent of the final presentation format. Its design allows to "just" use the functionality it provides, e.g., the access to media data, associated meta data, and user profile information, as well as the generation of the personalized multimedia content in standardized presentation formats. Hence, the framework relieves the developers of personalized multimedia applications from common tasks needed for content personalization, i.e., personalized content selection, composition functionality, and presentation generation, and lets them concentrate on their application-specific job. However, the framework is also designed to be extensible in regard of application-specific personalization functionality, for example by an application-specific personalized multimedia composition functionality. With the applications in the field of tourism and sports news we illustrated the usage of the framework in different domains and showed how the framework easily allows to dynamically create personalized multimedia content for different user contexts and devices.

The framework has been designed not to become yet another framework but to base on and integrate previous and existing research in the field. The design has been based on our long-term experience in advanced multimedia composition models and an extensive study of previous and ongoing related approaches. Its interfaces and extensibility explicitly allow not only to extend the framework's functionality but to embrace existing solutions of other (research) approaches in the field. The dynamically created personalized multimedia content needs semantically rich annotated content in the respective media data bases. In turn, the newly created content itself not only provides users with personalized multimedia information but at the same time forms a new, semantically even richer multimedia content that can be retrieved and reused. The composition operators provide common multimedia composition functionality but also allow the integration of very specific operators. The explicit decision for presentation-independence by a comprehensive internal composition model makes the framework both independent of any specific presentation format and prepares it for future formats to come.

Even though a software engineering approach towards dynamic creation of personalized multimedia content may not be the obvious one, we are convinced that our framework fills the gap between dynamic personalization support based on abstract data models, constraints or rules, and application-specific programming of personalized multimedia applications. With the MM4U framework, we are contributing to a new but obvious research challenge in the field of multimedia research, i.e., the shift from tools for the manual creation of static multimedia content towards techniques for the dynamic creation of, respectively, context-aware and personalized multimedia content, which is needed in many application fields. Due to its domain independence, MM4U can be used by arbitrary personalized multimedia applications, each application applying a different configuration of the framework. Consequently, for providers of applications the framework approach supports a cheaper and quicker development process and by this contributes to a more efficient personalized multimedia content engineering.

10. REFERENCES

3rd Generation Partnership Project (2003a). TS 26.234; Transparent end-to-end Packet-switched Streaming Service; Protocols and codecs (Release 5), 2003. Retrieved December 19, 2003, from http://www.3gpp.org/ftp/Specs/html-info/26234.htm

3rd Generation Partnership Project (2003b). TS 26.246; Transparent end-to-end Packet-switched Streaming Service; 3GPP SMIL language profile (Release 6), 2003. Retrieved December 19, 2003, from http://www.3gpp.org/ftp/Specs/html-info/26246.htm

Ackermann, P. (1996). Developing object oriented multimedia software: based on MET++ Application Framework. Heidelberg, Germany, dpunkt.

Adobe Systems, Inc., USA (2001). Adobe SVG Viewer, Retrieved February 25, 2004, from http://www.adobe.com/svg/

Allen, J. F. (1983, November). Maintaining knowledge about temporal intervals. In Commun. ACM. Volume 25, Number 11.

Agnihotri, L., Dimitrova, N., Kender, J., & Zimmerman, J. (2003). Music videos miner. In ACM Multimedia.

Amazon, Inc, USA (1996-2004). amazon.com. Retrieved February 20, 2004, from http://www.amazon.com/

Andersson, O., Axelsson, H., Armstrong, P., Balcisoy, S., et al. (2004a). Mobile SVG Profiles: SVG Tiny and SVG Basic. W3C Recommendation 25/03/2004. Retrieved June 10, 2004, from http://www.w3.org/TR/SVGMobile12/

Andersson, O., Axelsson, H., Armstrong, P., Balcisoy, S., et al. (2004b). Scalable Vector Graphics (SVG) 1.2 Specification. W3C Working Draft 05/10/2004. Retrieved June 10, 2004, from http://www.w3c.org/Graphics/SVG/

André, E., Müller, J., & Rist, T. (1996). WIP/PPP: Knowledge-Based Methods for Fully Automated Multimedia Authoring. In Proc. of EUROMEDIA'96. London, UK.

André, E., & Rist, T. (1995). Generating coherent presentations employing textual and visual material. In Artif. Intell. Rev. Volume 9, Number 2-3. Kluwer Academic Publishers.

André, E., & Rist, T. (1996, August). Coping with Temporal Constraints in Multimedia Presentation Planning. In *Proc. of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, Oregon.

Arndt, T. (1999, June). The Evolving Role of Software Engineering in the Production of Multimedia Applications. In IEEE International Conference on Multimedia Computing and Systems Volume 1, 1999, Florence, Italy.

Ayars, J., Bulterman, D., Cohen, A., Day, K., Hodge, E., Hoschka, P., et al. (2001). Synchronized Multimedia Integration Language (SMIL 2.0) Specification. W3C Recommandation 08/07/2001. Retrieved February 23, 2004, from http://www.w3c.org/AudioVideo/

Beckett, D., & McBride, B. (2003). RDF/XML Syntax Specification (Revised). W3C Recommendation 15/12/2003. Retrieved February 23, 2004, from http://www.w3c.org/RDF/

Bohrer, K., & Holland, B. (2004). Customer profile exchange (cpexchange) specification – version 1.0, 20/10/2000. Retrieved January 27, 2004, from http://www.cpexchange.org/standard/cpexchangev1_0F.zip

Boll, S. (2003, July). Vienna 4 U - WhatWeb Services can do for personalized multimedia applications. In Proc. of the Seventh Multi-Conference on Systemics Cybernetics and Informatics (SCI 2003), Orlando, Florida, USA.

Boll, S., & Klas, W. (2001). ZYX - A Multimedia Document Model for Reuse and Adaptation. In IEEE Transactions on Knowledge and Data Engineering, Volume 13, Issue 3.

Boll, S., Klas, W., Heinlein, C., & Westermann, U. (2001, August). Cardio-OP - Anatomy of a Multimedia Repository for Cardiac Surgery. Technical Report TR-2001301, University of Vienna, Austria.

Boll, S., Krösche, J., & Wegener, C. (2003, August). Paper chase revisited - a Real World Game meet Hypermedia (short paper). In Proc. of the Intl. Conference on Hypertext (HT03). Nottingham, United Kingdom.

Boll, S., Krösche, J., & Scherp, A. (2004, September). Personalized Multimedia meets Location-Based Services. In *Proc. of the "Multimedia-Informationssysteme" Workshop associated with the 34th annual meeting of the German Society of Computing Science*. Ulm, Germany.

Boll, S., Klas, W., & Wandel, J. (1999, November). A Cross-Media Adaptation Strategy for Multimedia Presentations, Proc. of the ACM Multimedia Conf. '99, Part 1, Orlando, Florida, USA.

Boll, S., Klas, W., & Westermann, U. (2000, August). Multimedia Document Formats - Sealed Fate or Setting Out for New Shores? In Multimedia - Tools and Applications. Number 11, Issue 3.

Boll, S., & Westermann, U. (2003, November). MediÆther - an Event Space for Context-Aware Multimedia Experiences. In Proc. of International ACM SIGMM Workshop on Experiential Telepresence, Berkeley, USA.

Bordegoni, M., Faconti, G., Feiner, S., Maybury, M. T., Rist, T., Ruggieri, S., Trahanias, P., & Wilson, M. (1997, December). A standard reference model for intelligent multimedia presentation systems. In ACM Computer Standards & Interfaces. Volume 18, Issue 6-7.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. In User Modeling and User Adapted Interaction. Volume 6, Issue 2-3.

Bulterman, D. C. A., van Rossum, G., & van Liere, R. (1991). A Structure of Transportable, Dynamic Multimedia Documents. In Proc. of the Summer 1991 USENIX Conf. Nashville, TN, USA.

Bugaj, S., Bulterman, D., Butterfield, B., Chang, W., Fouquet, G., Gran, C., et al. (1998). Synchronized Multimedia Integration Language (SMIL 1.0) Specification. W3C Recommandation 06/15/1998. Retrieved June 10, 2004, from http://www.w3.org/TR/REC-smil/

Chen, G., & Kotz, D. (2000). A Survey of Context-Aware Mobile Computing Research. Dartmouth Computer Science, Technical Report TR2000-381.

Click2learn, Inc, USA (2001-2002). Toolbook – Standards-Based Content Authoring. Retrieved February 6, 2004, from http://home.click2learn.com/en/toolbook/index.asp

CWI (2004). The Cuypers Multimedia Transformation Engine. Amsterdam, The Netherlands. Retrieved February 25, 2004, from http://media.cwi.nl:8080/demo/.

De Bra, P., Brusilovsky, P., & Houben, G.-J. (1999a, December). Adaptive Hypermedia: From Systems to Framework. In ACM Computing Surveys. Volume 31, Issue 4.

De Bra, P., Houben, G.-J., & Wu, H. (1999b). AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In Proc. of the tenth ACM Conf. on Hypertext and hypermedia: returning to our diverse roots. Darmstadt, Germany.

De Bra, P., Aerts, A., Houben, G.-J., & Wu, H. (2000). Making General-Purpose Adaptive Hypermedia Work. In Proc. of the AACE WebNet Conference. San Antonio, Texas.

De Bra, P., Aerts, A., Smits, D., & Stash, N. (2002a, October). AHA! Version 2.0: More Adaptation Flexibility for Authors. In Proc. of the AACE ELearn'2002 Conf.

De Bra, P., Brusilovsky, P., & Conejo, R. (2002b, May). Proc. of the Second Intl. Conf. for Adaptive Hypermedia and Adaptive Web-Based Systems, Malaga, Spain, Springer LNCS 2347.

De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D., & Stash, N. (2003, August). AHA! The Adaptive Hypermedia Architecture. Proceedings of the ACM Hypertext Conference, Nottingham, UK.

De Carolis, B., de Rosis, F., Andreoli, C., Cavallo, V., De Cicco, M L (1998). The Dynamic Generation of Hypertext Presentations of Medical Guidelines. In The New Review of Hypermedia and Multimedia. Volume 4.

De Carolis, B., de Rosis, F., Berry, D., & Michas, I. (1999). Evaluating plan-based hypermedia generation. In Proc. of European Workshop on Natural Language Generation. Toulouse, France.

de Rosis, F., De Carolis, B., & Pizzutilo, S. (1999). Software documentation with Animated Agents. In Proc. of the 5th ERCIM Workshop on User Interfaces For All. Dagstuhl, Germanny.

Dublin Core Metadata Initiative (1995-2003). Expressing Sinple Dublin Core in RDF/XML,1995-2003. Retrieved February 2, 2004, from http://dublincore.org/documents/2002/07/31/dcmes-xml/

Duke, D. J., Herman, I., & Marshall, M. S. (1999). PREMO: A Framework for Multimedia Middleware: specification, rationale, and java binding, Springer.

Duda, A., & Keramane, C. (1995). Structured Temporal Composition of Multimedia Data. Proceedings of the IEEE International Workshop Multimedia-Database-Management Systems.

Echiffre, M., Marchisio, C., Marchisio, P., Panicciari, P., & Del Rossi, S. (1998, January-March). MHEG-5 – Aims, Concepts, and Implementation Issues. In IEEE Multimedia.

Egenhofer, M. J., & Franzosa, R. (1991, March). Point-Set Topological Spatial Relations. In Int. Journal of Geographic Information Systems. Number 5, Issue 2.

Elhadad, M., Feiner, S., McKeown, K., & Seligmann, D. (1991). Generating customized text and graphics in the COMET explanation testbed. In *Proc. of the 23rd conf. on Winter simulation*. IEEE Computer Society. Phoenix, Arizona, USA.

Engels, G., Sauer, S., & Neu, B. (2003, October). Integrating Software Engineering and User-centred Design for Multimedia Software Developments. In Proc. IEEE Symposia on Human-Centric Computing Languages and Environments - Symposium on Visual/Multimedia Software Engineering. Auckland, New Zealand. IEEE Computer Society Press.

Exor International Inc. (2001-2004). eSVG: Embedded SVG. Retrieved February 12, 2004, from http://www.embedding.net/eSVG/english/overview/overview_frame.html

Fink, J., Kobsa, A., & Schreck, J. (1997). Personalized hypermedia information through adaptive and adaptable system features: User modeling, privacy and security issues. In Intelligence in Services and Networks: Technology for Cooperative Competition, A. Mullery, M. Besson, M. Campolargo, R. Gobbi, and R. Reed, Eds. Springer, Berlin, Heidelberg.

Foundation for Intelligent Physical Agents (2002). FIPA Device Ontology Specification, 2002. Retrieved January 23, 2004, from http://www.fipa.org/specs/fipa00091/

Gaggi, O., & Celentano, A. (2002). A Visual Authoring Environment for Prototyping Multimedia Presentations. In Proceedings of the IEEE Fourth International Symposium on Multimedia Software Engineering.

Girgensohn, A., Bly, S., Shipman, F., Boreczky, J., & Wilcox, L. (2001). Home Video Editing Made Easy – Balancing Automation and User Control. In Proc. of the Human-Computer Interaction. Tokyo, Japan.

Girgensohn, A., Shipman, F., & Wilcox, L. (2003, November). Hyper-Hitchcock: Authoring Interactive Videos and Generating Interactive Summaries. In Proc. ACM Multimedia.

Greiner, C., & Rose, T. (1998, November). A Web Based Training System for Cardiac Surgery: The Role of Knowledge Management for Interlinking Information Items. In Proc. The World Congress on the Internet in Medicine. London, UK.

Hardman, L. (1998, March). Modeling and Authoring Hypermedia Documents. PhD thesis, University of Amsterdam, Amsterdam, Netherlands.

Hardman, L., van Rossum, G., Jansen, J., & Mullender, S. (1994a). CMIFed: a transportable hypermedia authoring system. In Proc. of the second ACM international conference on Multimedia. San Francisco, California, USA.

Hardman, L., Bulterman, D. C. A., & van Rossum, G. (1994b, February). The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. In Comm. of the ACM. Number 37, Issue 2.

Hirzalla, N., Falchuk, B., & Karmouch, A. (1995). A Temporal Model for Interactive Multimedia Scenarios. In IEEE Multimedia. Number 2, Issue 3.

Hunter, J. (1999, October). Multimedia Metadata Schemas. Retrieved June 16, 2004, from http://www2.lib.unb.ca/Imaging_docs/IC/schemas.html

International Organisation for Standardization (1996). ISO 13522-5, Information Technology – Coding of Multimedia and Hypermedia Information, Part 5: Support for Base-Level Interactive Applications. Int'l Org. for Standardization, Geneva.

Jourdan, M., Layaïda, N., Roisin, C., Sabry-Ismaïl, L., & Tardif, L. (1998). Madeus, and Authoring Environment for Interactive Multimedia Documents. ACM Multimedia.

IBM Corporation, USA. (2004a). IBM Research – Video Semantic Summarization Systems. Retrieved June 15, 2004 from http://www.research.ibm.com/MediaStar/VideoSystem.html#Summarization%20Techniques

IBM Corporation, USA. (2004b). QBIC Home Page. Retrieved June 16, 2004 from http://wwwqbic.almaden.ibm.com/

INRIA Rhône-Alpes, France (2003). PocketSMIL 2.0. Retrieved February 24, 2004, from http://opera.inrialpes.fr/pocketsmil/

ISO/IEC (Int. Organization for Standardization/Int. Electrotechnical Commission) JTC 1/SC 29/WG 11. (1999, July). MPEG-7: Context, Objectives and Technical Roadmap, V.12. ISO/IEC Document N2861. Geneva, Switzerland.

ISO/IEC JTC 1/SC 29/WG 11. (2001a, November). InformationTechnology—Multimedia Content Description Interface—Part 1: Systems. ISO/IEC Final Draft International Standard 15938-1:2001. Geneva, Switzerland.

ISO/IEC JTC 1/SC 29/WG 11. (2001b, September). Information Technology—Multimedia Content Description Interface—Part 2: Description Definition Language. ISO/IEC Final Draft Int. Standard 15938-2:2001. Geneva, Switzerland.

ISO/IEC JTC 1/SC 29/WG 11. (2001c, July). Information Technology—Multimedia Content Description Interface—Part 3: Visual. ISO/IEC Final Draft Int. Standard 15938-3:2001. Geneva, Switzerland.

ISO/IEC JTC 1/SC 29/WG 11. (2001d, June). Information Technology—Multimedia Content Description Interface—Part 4: Audio. ISO/IEC Final Draft Int. Standard 15938-4:2001. Geneva, Switzerland.

ISO/IEC JTC 1/SC 29/WG 11. (2001e, October). Information Technology—Multimedia Content Description Interface—Part 5: Multimedia Description Schemes. ISO/IEC Final Draft Int. Standard 15938-5:2001. Geneva, Switzerland.

Kim, M., Wood, S., & Cheok, L.-T. (2000, November). Extensible MPEG-4 Textual Format (XMT). In Proc. of the 8th ACM Multimedia Conf. Los Angeles, California, USA.

Klas, W., Greiner, C., & Friedl, R. (1999, July). Cardio-OP: Gallery of Cardiac Surgery. IEEE International Conference on Multimedia Computing and Systems (ICMS 99). Florence, July.

Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M. H., & Tran, L. (2003). Composite Capability/Preference Profile (CC/PP): Structure and Vocabularies - W3C Working Draft 25/03/2003.

Kopf, S., Haenselmann, T., Farin, D., & Effelsberg, W. (2004). Automatic Generation of Summaries for the Web. In: Proc. Electronic Imaging 2004.

Lemlouma, T., & Layaïda, N. (2003, June). Media Resources Adaptation for Limited Devices. In Proc. of the 7th ICCC/IFIP International Conference on Electronic Publishing ELPUB 2003. Universidade deo Minho, Portugal.

Lemlouma, T., & Layaïda, N. (2004, January). Context-Aware Adaptation for Mobile Devices. IEEE International Conference on Mobile Data Management. Berkeley, CA, USA.

Little, T. D. C., & Ghafoor, A. (1993). Interval-Based Conceptual Models for Time-Dependent Multimedia Data. In IEEE Transactions on Knowledge and Data Engineering. Volume 5, Number 4.

Macromedia, Inc., USA (2003, January). Using Authorware 7 [Computer manual]. Available from http://www.macromedia.com/software/authorware/

Macromedia, Inc., USA (2004). Macromedia. Retrieved June 15, 2004, from http://www.macromedia.com/

McKeown, K., Robin, J., & Tanenblatt, M. (1993). Tailoring lexical choice to the user's vocabulary in multimedia explanation generation. In Proc. of the 31st conference on Association for Computational Linguistics. Columbus, Ohio.

Oldenettel, F., & Malachinski, M. (2003, May). The LEBONED Metadata Architecture. In Proc. of the 12th International World Wide Web Conference. Budapest, Hungary, ACM Press, S.207-216, Special Track on Education, 2003

Open Mobile Alliance (2003). User Agent Profile (UA Prof). – 20/05/2003. Retrieved February 10, 2004, from http://www.openmobilealliance.org/

Oratrix, Netherlands (2004). GRiNS for SMIL Homepage. Retrieved February 23, 2004, from http://www.oratrix.com/GRiNS

Papadias, D., & Sellis, T. (1994, October). Qualitative Representation of Spatial Knowledge in Two-Dimensional Space. In VLDB Journal. Number 3, Issue 4.

Papadias, D., Theodoridis, Y., Sellis, T., & Egenhofer, M. J. (1995, March). Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-Trees. In Proc. of the ACM SIGMOD Conf. on Management of Data, San Jose.

Pree, W. (1995). Design Patterns for Object-Oriented Software Development. Addison-Wesley, 1995.

Rabin, M. D., & Burns, M. J. (1996). Multimedia authoring tools. In Conference companion on Human factors in computing systems, Vancouver, British Columbia, Canada, ACM Press.

Raggett, D., Le Hors, A., & Jacobs, I. (1998). HyperText Markup Language (HTML) version 4.0. W3C Recommendation, revised on 04/24/1998. Retrieved February 20, 2004, from http://www.w3c.org/MarkUp/

RealNetworks, USA (2003). RealOne Player. Retrieved February 25, 2004, from http://www.real.com/

Rossi, G., Schwabe, D., & Guimarães, R. (2001, May). Designing Personalized Web Applications, 2001. In Proceedings of the tenth World Wide Web (WWW) Conference, ACM. Hong Kong.

Rout, T. P., & Sherwood, C. (1999, May). Software Engineering Standards and the Development of Multimedia-Based Systems. In Fourth IEEE International Symposium and Forum on Software Engineering Standards. Curitiba, Brazil.

Scherp, A., & Boll, S. (2004a, March). mobileMM4U - framework support for dynamic personalized multimedia content on mobile systems, In Multikonferenz Wirtschaftsinformatik 2004, special track on Technologies and Applications for Mobile Commerce.

Scherp, A., & Boll, S. (2004b, October). Generic Support for Personalized Mobile Multimedia Tourist Applications. Technical Demonstration for the ACM Multimedia Conference. New York, USA.

Scherp, A. & Boll, S. (2005, January). Paving the Last Mile for Multi-Channel Multimedia Presentation Generation. In *Proc. of the 11th International Conference on Multimedia Modeling*, Melbourne, Australia.

Schmitz, P., Yu, J., & Santangeli, P. (1998). Timed Interactive Multimedia Extensions for HTML (HTML+TIME). W3C, version 09/18/1998. Retrieved February 20, 2004, from http://www.w3.org/TR/NOTE-HTMLplusTIME

Sun Microsystems, Inc., USA (2004). Java Media Framework API. Retrieved February 15, 2004, from http://java.sun.com/products/java-media/jmf/index.jsp

Stash, N., & De Bra, P. (2003, June). Building Adaptive Presentations with AHA! 2.0. Proceedings of the PEG Conference, Sint Petersburg, Russia.

Stash, N., Cristea, A., & De Bra, P. (2004). Authoring of Learning Styles in Adaptive Hypermedia. In WWW’04 Education Track, ACM. NY, USA.

Szyperski, C., Gruntz, D., & Murer, S. (2002). Component software: beyond object-oriented programming, 2nd edition, Addison-Wesley.

Yahoo!, Inc., USA (2002). MyYahoo!. Retrieved February 17, 2004, from http://my.yahoo.com/

van Ossenbruggen, J.R., Cornelissen, F.J., Geurts, J.P.T.M., Rutledge, L.W., & Hardman, H.L. (2000, December). Cuypers: a semi-automatic hypermedia presentation system. Technical Report INS-R0025. CWI, Netherlands.

van Rossum, G., Jansen, J., Mullender, S., & Bulterman, D. C. A. (1993). CMIFed: a presentation environment for portable hypermedia documents. In Proc. of the first ACM international conference on Multimedia. Anaheim, California, USA.

Villard, L. (2001, November). Authoring Transformations by direct manipulation for Adaptable Multimedia Presentations, In Proceeding of the ACM Symposium on Document Engineering. Atlanta, Georgia.

Wahl, T., & Rothermel, K. (1994, May). Representing Time in Multimedia Systems. In Proc. IEEE Int. Conf. on Multimedia Computing and Systems. Boston, MA, USA.

Wu, H., de Kort, E., & De Bra, P. (2001). Design Issues for General-Purpose Adaptive Hypermedia Systems. In Proc. of the twelfth ACM Conf. on Hypertext and Hypermedia. Århus, none, Denmark.