12 ZyX — A SEMANTIC MODEL FOR MULTIMEDIA DOCUMENTS AND PRESENTATIONS

Susanne Boll, Wolfgang Klas

Database and Information Systems (DBIS)
University of Ulm, Computer Science Department, Ulm, Germany
{boll,klas}@informatik.uni-ulm.de

Abstract: Existing languages, formats, and multimedia document models such as HTML, MHEG, SMIL, HyTime, SGML, and XML, do not provide the appropriate modeling primitives needed to provide adequate support for *reusability*, *interaction*, *adaptation*, and *presentation-neutral description* of the structure and content of multimedia documents as required in the Cardio-OP project. Since each of these models lacks some significant concepts and does not meet all of the requirements, we propose a new approach for the semantic modeling of multimedia content, the ZyX model, which we implemented on the basis of an object-relational database system. The approach taken allows for fine-grained representation and retrieval of structures and layout of multimedia material, for flexible on-the-fly composition of multimedia fragments in order to create individualized multimedia documents, and for the realization of adaptation and personalization of multimedia presentations depending on the user environment specified by means of user profiles.

12.1 INTRODUCTION

Application fields that make dedicated use of various types of media most often require a well-organized and sound representation of the many semantic components and relationships defined between the different types of media. Depending on the requirements of such applications, one can choose from a variety of "data models" for modeling multimedia content, e.g., proprietary formats given by commercial products, or standards like HTML, MHEG [11, 10, 9], SMIL [6], and document models constructed by applying HyTime [8, 15], SGML [7], or XML [4]. All of these formats are quite open with regard to the formats of

the "atomic" constituents like MPEG or JPEG they allow to build on. However, the various formats and document models differ significantly in various aspects: support for semantic modeling, interaction, reusability, flexible composition, adaptation and individualization for presentation, presentation-neutral storage, and Internet applicability. This is mainly due to the fact that the development of these formats took place in viewpoint of particular types of applications. HTML was developed for the WWW, which basically follows the hypertext approach, i.e., an interlinking of individual pages which do not form an integrated multimedia presentation. The development of MHEG-5 was mainly driven by the set-top-box type of application which resulted in a model that tries to organize the world by means of a collection of interconnected scenes. The HyTime development was driven by the idea of extending SGML toward support for multimedia documents, but it was based on a notion of documents which does not include user interaction. SMIL [6] has been developed for synchronized multimedia presentations in the Internet.

Background for our work is the project "Gallery of Cardiac Surgery" (CardioOP[1]). The overall goal is to develop a network-based and database-driven multimedia information system for physicians, medical lecturers, students, and patients in the domain of cardiac surgery. The system will serve as a common information and education base for its different types of users in the domain of cardiac surgery. The users are provided with multimedia information according to their specific request, their different understanding of the selected subject, their geographic location and technical infrastructure.

Within this project context, our group is developing concepts and prototypical implementations of a database-driven multimedia repository that integrates *modeling*, *management*, and *content-based retrieval* of multimedia content with flexible dynamic multimedia presentation services that *deliver* and *present* the multimedia content according to the user context. Major project requirements are the support for *interaction*, *reusability*, *adaptation*, and *presentation-neutral description* of the structure and content of multimedia documents.

As the information system is shared by different user groups it will be designed to support flexible re-use of the multimedia material in different context, with different communication media (on-line, CD-ROM, print media), and at different locations (university campus, hospitals, at home). The repository will contain pre-orchestrated reusable multimedia document fragments which can be composed on-the-fly to final multimedia documents. This calls for a presentation-neutral representation of multimedia content in the database allowing to store modular multimedia documents independent of the final presentation format. The quality of a presentation still is a parameter for such a

[1]Partially funded by the German Ministry of Research and Education, grant number 08C58456. Our project partners are the University Hospital of Ulm, Dept. of Cardiac Surgery and Dept. of Cardiology, the University Hospital of Heidelberg, Dept. of Cardiac Surgery, an associated Rehabilitation Hospital, the publishers Barth-Verlag and dpunkt-Verlag, Heidelberg, FAW Ulm, and ENTEC GmbH, St. Augustin. For details see also URL www.informatik.uni-ulm.de/dbis/GH/

composition of multimedia document fragment and will be determined by the output channel chosen by the user, e.g., at the university campus or at home.

Given these requirements, we face serious problems concerning the support for modeling the content given in the project by existing document models based on formats like HTML, MHEG, SMIL, HyTime, SGML/DSSSL, or XML. In this paper, we present the ZyX model, which forms the core for the modeling of the multimedia data in our repository. In comparison to existing models, it provides more adequate support for *semantic modeling*, *interaction*, *reusability* and *flexible composition*, *adaptation* and *individualization* for presentation, *presentation-neutral storage*, and *Internet applicability*.

The paper is organized as follows: Section 12.2 first provides a better understanding of the requirements, which leads to a metric that we used to analyze existing models. Second, the result of the analysis is summarized. Section 12.3 presents the basic ideas and design considerations of our model and gives a formal framework for a more detailed understanding. Section 12.4 summarizes our work and gives an outlook to future work.

12.2 BASIC REQUIREMENTS AND ANALYSIS OF EXISTING MODELS

In order to support modular and context-dependent composition of multimedia documents from media objects and parts of multimedia documents, we have to provide a data model which meets the following criteria:

Reusability. Reusability of document components should be supported along three dimensions: (1) *granularity* of the components, i.e., reuse of complete multimedia documents, fragments of multimedia documents, or individual atomic media objects, (2) *kind* of re-usage, i.e., identical reusage including all temporal, spatial, design and interaction relationships as given by the author, or structural reusage by means of separating layout and structure and reusing only structural parts, and (3) *selection* and *identification* of components, which calls for mechanisms for classifying, indexing, and querying components.

Interaction. Cardio-OP users should be able to interact with presentations in terms of three types of interaction: (1) *Navigational interactions* determining the user-defined flow of a multimedia presentation, (2) *design interactions* influencing the visual and audible layout of a presentation, and (3) *movie interactions* affecting the temporal course of the *entire* presentation. Navigational and design interactions should be specified within multimedia documents, whereas movie interactions are expected to be offered by the presentation engine.

Adaptation. Cardio-OP presentations should be adaptable to *user-specific characteristics*, i.e., personal interest of a user by means of professional level, degree of details, or user preferences, and by means of a user's technical infrastructure like kind of network connection, on-campus or off-campus locations. Adaptation of multimedia presentations should take place at the *client* and/or at the *server*, whatever is most suitable for the kind of adaptation needed.

Presentation-neutral Representation. The multimedia material available in the Cardio-OP repository has to be presentable in a heterogeneous software

and hardware environment. As a consequence, the multimedia material has to be stored presentation-neutral, i.e., independent of the actual realization of a presentation at a client. This calls for converting a presentation-neutral representation of multimedia content into a presentation-specific format used for playout of the multimedia material. It is desirable that this conversion is lossless. The presentation-neutral representation of multimedia content should — besides the coverage of rich multimedia functionality — take place on a high level of semantics. The presentation-neutral model should be open in the sense that it allows for later integration of multimedia functionality expected to be developed in the future.

Figure 12.1 summarizes the analysis of the most relevant existing approaches and shows to which extent MHEG-5/6, HyTime, and SMIL, fulfill the specific requirements. Due to the limitation of space we can not present the comprehensive discussion how they meet the specific requirements in this paper but refer the reader to specific literature for a detailed description of the standards and data models. The analysis of existing standards, defacto standard formats, and

	MHEG-5 / MHEG-6	SMIL	HyTime
Reusage			
Granularity			
atomic media objects	+	+	+
document parts	-	-	+
complete documents	+	+	+
Kind of re-usage			
identical	+	+	+
structural	-	-	+
Identification/Selection	-	o	+
Interaction			
Navigational interactions	+	+	-
Design interactions	+	-	-
Adaptation			
User-specific Adaptation			
to personal interest	-/+	-	-
to technical infrastructure	-/+	o	-
Location of Adaptation			
Server-based	-	-	-
Client-based	-/+	+	-
Presentation-neutral representation			
Presentation-neutral	-	-	+
High semantic level	-	o	+

Figure 12.1: Summary of the support of the requirements by MHEG-5/6, SMIL, and HyTime (+ support, o partial support, — no support)

models shows that, although, individual formats and models are strong with

respect to particular features, they are not capable to meet all the requirements identified in the Cardio-OP project. This led to the design and implementation of the ZyX model which tries to exploit the features of existing formats and models, especially also recent developments in the area of Internet-applicable models driven by the development of XML and SMIL.

12.3 THE ZyX MODEL

In the following, we first sketch some design considerations of our model and the points of contact with other approaches in the field in Section 12.3.1. In Section 12.3.2, we introduce the reader into the basic concepts of our ZyX data model before we present a formal framework in Section 12.3.3.

12.3.1 Design Considerations

For the design of the new model we considered the semantic level of the data model, the underlying temporal and spatial model, interaction capabilities, adaptation modeling and presentation neutral representation.

Semantic level. The semantic level of the data model is important for the support of both reusability of multimedia documents and fragments of multimedia documents and presentation-neutral representation of multimedia documents. We decided to develop a data model that describes a multimedia document on a high semantic level. This allows us a (lossy) *export* of our multimedia documents into data models like MHEG-5, SMIL, and HTML.

Document structure. For the structure of the document we consider a hierarchical organization of the document as can be found with SMIL documents. However, the modeling capabilities of our data model extends those of SMIL by the aspects of reusability and the possibilities for modeling adaptation to users interests.

Temporal model. We decided to use an interval-based temporal model. One important requirement to the temporal model hereby is its capability to describe the temporal dimension of interaction. Existing interval-based temporal models are mainly based on some or all of the 13 binary temporal relations between time intervals as defined by Allen [1]. These models, however, do not support time intervals of unknown duration that occur, e.g., in the context of user interaction in multimedia presentations (e.g., Object Composition Petri Nets (OCPN) [13]). With the Interval Expressions [5] we find a temporal model for multimedia presentations on the level of intervals with a set of temporal operators to relate time intervals which possibly have an unknown duration, that also overcomes the problem of temporal inconsistencies by construction. The Interval Expressions form the basis of the underlying temporal model of the ZyX data model.

Spatial model. We constrain ourselves to a simple spatial model as we emphasize the modeling of the temporal course, interaction, adaptation and reusability with our ZyX model. We decided to support the spatial layout by a point-based description of each visual media entity in a multimedia document. Each visual

media entity has assigned 2-dimensional extension plus a third dimension to specify overlapping of visual media entities. We do not consider the specification of spatial relationships between media entities like *right-of* or *besides*. The ZyX data model, however, allows to be extended by a more sophisticated spatial model later.

Interaction. Our model supports the two interaction types navigational/ decision interactions and design interactions. This means that our model provides a comprehensive support for these two interaction types comparable with the interaction capabilities of MHEG-5, but more sophisticated than HyTime and SMIL.

Adaptation. Our model supports adaptation mechanisms like can be found with SMIL but that go far beyond the adaptation capabilities of SMIL. In SMIL adaptation is limited to the exploitation of a set of discriminating attributes of a `switch` element, like system-bitrate and system-language to select one out of a set of alternatives by evaluating these attributes. Our support for adaptation within the data model of the multimedia document is twofold — at selection time of the document and at presentation time of the document. Adaptation at selection time means that the document itself is adjustable to the user's interest and system environment before it is executed for presentation. Adaptation at presentation time means that in a client/server environment the presentation engine exploits information of the multimedia document to adjust, e.g., the quality of the selected media elements to cope with a lower network bandwidth.

Presentation-neutral representation. The presentation neutral representation of multimedia documents is strongly related to our requirement of reusability at different levels of granularity. We developed a generic representation that comprises the different granules: media elements, document fragment, and entire documents. The design allows to reuse and to compose these building block in an arbitrary fashion.

12.3.2 Basic Concepts of the ZyX Model

In this section, we present the terminology and the basic concepts of the ZyX model. The ZyX model describes a multimedia document by means of a tree. The nodes of the tree are the *presentation elements* and the edges of the tree *bind* the presentation elements together in a hierarchical fashion. Each presentation element has one *binding point* with which it can be bound to another presentation element to it. It also has one or more *variables* with which it can bind other presentation elements. Figure 12.2 shows the graphical representation of these basic elements of the model.

Presentation elements are the generic elements of the model. They can be mere media elements or hold the place for *fragments*. They can also be elements that represent the temporal, spatial, layout, and interactive semantic relationships between the elements of a multimedia document. Consider the example in Figure 12.3. A temporal element, e.g., the sequential element *seq*, binds the media elements $slide_1$, $slide_2$ and $slide_3$ to its variables v_2, v_3, and

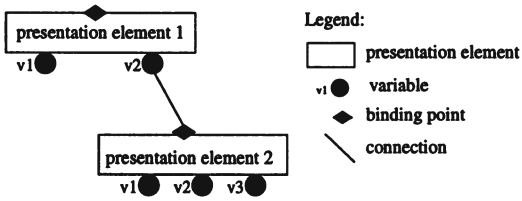

Figure 12.2: Graphical representation of the basic document elements

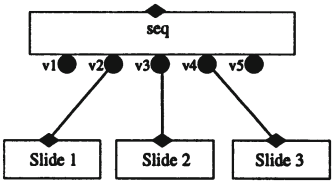

Figure 12.3: Simple document tree with seq element

v_4. This represents the semantic relationship of the sequential presentation of the three slides. With the *seq* element's binding point this sequential slide presentation can be bound to another presentation element in a more complex multimedia document tree.

We now explain the modeling capabilities of our model with regard to our specific requirements of reusability, interaction, adaptation and presentation-neutral representation.

Reusability. In the ZγX model, not all of the variables of a presentation element must be bound at authoring time. In Figure 12.3 the variables v_1 and v_5, e.g., the title and the summary of the slide presentation are still unbound. This means that the slide sequence can later be completed by binding presentation elements to the *free variables*. The simple sample tree in Figure 12.3 hence forms a "template". This is an important feature for building reusable *fragments* that can be reused in different multimedia documents by binding the free variables differently corresponding to the context.

It is also possible to form more complex fragments like the one shown in Figure 12.4. Here, on different "levels" of the specification tree variables are left unbound. To make later composition of such fragments easier a fragment can be *encapsulated* by a *complex media element*. This means that a fragment appears like a single presentation element in the specification tree with one binding point and a set of free variables. The free variables of the fragment are *exported*. Figure 12.4 illustrates how a complex media element encapsulates a complex fragment. The complex media element somehow is the black box view to a complex presentation fragment. Analogously, an *external media element* encapsulates a specification of a fragment that was composed in another *external* document format. This allows that the inclusion of existing documents into our model. What, however, is encapsulated by the external media element is dependent of the external document format. The concepts of free variables and complex media element guarantee reusability on the level of presentation fragments. External media elements allow for document format comprehensive reusability.

Reusability on the level of media elements is supported by means of *selector elements*. These are presentation elements that determine *what*, that is which

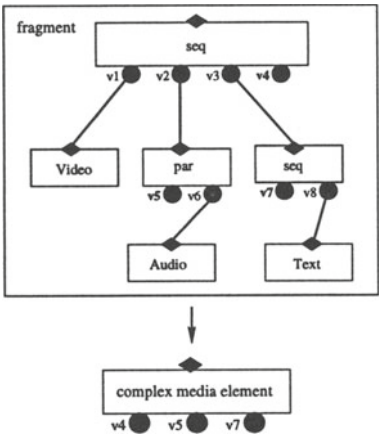

Figure 12.4: Complex fragment encapsulated in a complex media element

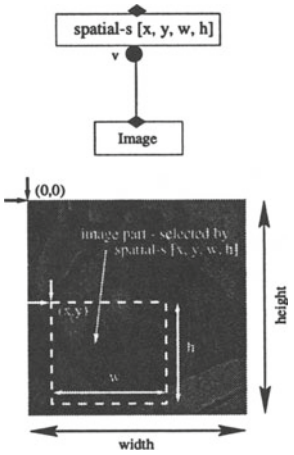

Figure 12.5: Spatial selector element — *spatial-s* $[x, y, w, h]$ and its semantics

part of a media element is presented. They can be used to select a specific part of an audio or a specific area of an image. Figure 12.5 illustrate such a usage and semantics of a spatial selector element. Here, a spatial selector is applied to an image media element to select a rectangular area from the image. The selectors can be applied to both media elements as well as to fragments, e.g., a temporal selector to select two minutes of an existing slide presentation. The selectors can be organized in a hierarchy so that, e.g., one can select a part of a video element with a certain duration by means of a temporal selector and use only a specific detail by means of a spatial selector.

Besides the selector elements the ZyX data model offers *projector elements* that influence the visual and audible layout in a presentation of a multimedia document. Projector elements determine *how* a media element or a fragment is presented. They determine for example the presentation speed of a video or the spatial position of a video on the screen. Projectors can only be bound to *projector variables* of media elements and fragments. Each presentation element can have one or more projector variables to which projectors can be bound. Figure 12.6 illustrates the usage of projector elements and their semantics when the document is presented. In this example a fragment defines the sequential presentation of two text elements and a video. Two projector elements are bound to the sequential element, a spatial projector and a typographic projector. A projector applies not only to the presentation element it is bound to but also to its subtree. However, a projector applies only to those elements in the same tree that can be affected by it. A spatial projector affects the layout of an image but not that of an audio. The semantics of the spatial projector is to define the position of all three visual elements for a presentation. The typographic projector applies to the two text elements and sets

their font, size and style for presentation. The appearance of an audio by an acoustic projector element and so on. By means of the projector elements one can add two layouts to the same document. This allows for reusability of the same document in different presentation contexts.

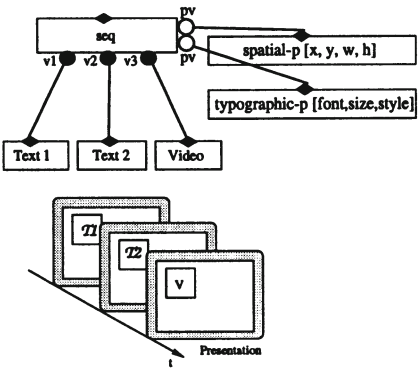

Figure 12.6: Simple fragment with spatial and typographic projector elements — *spatial-p* $[x, y, w, h]$ *typographic-p* $[font, size, style]$ and their semantics

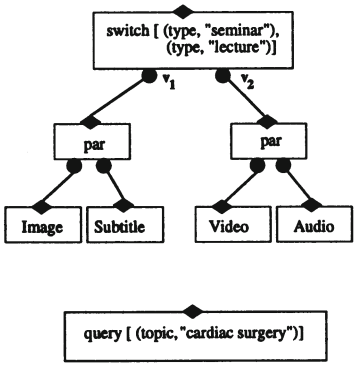

Figure 12.7: Specification of presentation alternatives with the switch and the query element

Adaptation. Adaption in the context of the ZyX model means that the multimedia document delivered to a user for presentation in respond to a user query should match the user's interest and the user's system environment collected in a user *profile*. Therefore, each fragment is assigned a set of meta data that describes its content. In addition to the multimedia document a user profile, also meta data, is defined to capture values that describe a user, topics of interest, presentation system environment, network connection characteristics and the like. *Meta data* that describes users and fragments is organized as key-value pairs.

The ZyX data model offers presentation elements to support adaptation to a user's profile by means of *switch elements* and the *query elements*.

The switch elements allows to specify different alternatives for a part of the document. One of the alternatives is selected corresponding to the user profile. An illustration of the switch element is given in Figure 12.7. When the document is presented depending of the type of teaching either the left or the right subtree is presented. A switch element is used if all alternatives are known to the author of the fragment.

However, there might be the case that the selected alternative can be determined only at selection time. For example, an author wants to determine that at a specific point in the presentation about "cardiac surgery" a digression to physiology is to be made but does not specify which fragments are relevant

to this. This can be specified with a *query* element. The query is represented by a set of meta data. When the document is selected for presentation the query element is evaluated and replaced by the fragment best matching the meta data. An illustration of the query element is given in Figure 12.7. The sample query element is the place holder for the fragment best matching the query with topic "cardiac surgery". The more meta data tuples are used the more specific the query is.

Interaction. The requirement to support the modeling of interactive multimedia presentations is met by the data model's *interaction elements.* The model offers two types of interaction elements, *navigational interactive elements* and *design interactive elements.* Example for a navigational elements is the *link* element that allows to specify hypertext structure. A *menu* element supports to interactively follow of one presentation out of a set of presentations paths. The design interactive elements are the interactive version of the projector elements. For example, for the typographic projector that allows to specify font, size and style of a text, the *interactive typographic selector element* specifies that these settings can be carried out interactively when the document is presented.

Presentation-neutral representation. The usage of projectors does not mean that the layout is statically anchored in the document. As outlined before not all variables of a presentation element must be bound in the first place. They can be bound when the document is selected for presentation. This is also the point in time when the projector variables of a document can be bound to a set of projectors. This follows the idea of separating structure from layout information as can be found with SGML and XML and complies with our requirement for presentation-neutral representation of the documents.

12.3.3 Formal Framework of the ZyX Model

12.3.3.1 Basic Terminology. The *presentation elements* are the generic elements of the ZyX model. Each presentation element p has assigned exactly one *binding point* b_p. This is the connector with which a presentation element can be bound to another presentation element. A presentation element has furthermore 0 to n *variables* v which are used to bind other presentation elements to it. To add layout information to a presentation element it optionally can have 0 to n *projector variables* pv that can be used to bind *projector elements* to the element. Projector variables can be seen just as "normal" variables, that is $v \equiv pv$. The projector variables are separated due to separating structure and layout (see Section 12.3.3.3). In the following definitions, let denote B the set of all binding points, VAR the set of all variables, $PVAR$ the set of all projector variables, T the set of all element types, MT the set of media types, M the set of all raw media data, $OT \subseteq T$ the set of all operator element types. A presentation element p can therefore be defined as follows:

Definition 12.3.1 (Presentation element)
A presentation element p is a tuple $p : [t_p, b_p, V_p, PV_p]$ with $t_p \in T$ denoting the type of p, $b_p \in B$ denoting the binding point of p, $V_p \subseteq VAR$ denoting the set of variables of p, and $PV_p \subseteq PVAR$ denoting the set of projector variables of p. p can be augmented with further tuple elements depending on its type t_p.

A presentation element p can be an atomic media element, a complex media element, an external media element, or a specific element to build up the temporal, structural and interactive relationships of a multimedia presentation.

The basic units of a multimedia document are the *atomic media elements.* An atomic media element is an instantiation of a *media type.* The atomic media element in our model abstracts from the raw media data and just represents the media element and its media specific characteristics.

Definition 12.3.2 (Atomic media element)
An atomic media element $am : [t_{am}, b_{am}, V_{am}, PV_{am}, m]$ is a presentation element with $t_{am} \in MT = \{Audio, Video, Image, Text\} \subseteq T$, $V_{am} = \emptyset$, and $m \in M$ denoting the media data represented by am.

Presentation elements are interconnected via the variables and binding points. In the graphical representation connections are represented by edges between presentation elements (see Figure 12.2). A *connection* is defined as:

Definition 12.3.3 (Connection)
A connection $c = [v, b_{p'}]$ connects the (projector) variable $v \in V_p \cup PV_p$ of a presentation element p with the binding point $b_{p'}$ of presentation element $p' \neq p$.

The result of interconnecting presentation elements is a specification tree that describes a *fragment.* A fragment encapsulates a reusable part of a multimedia document be it a single media element, a part, or an entire multimedia document. The formal description of a valid fragment is given in Definition 12.3.4.

Definition 12.3.4 (Fragment)
A fragment $f = (P, C)$ is an acyclic, undirected graph that describes a part or an entire multimedia document with:

- *P the set of presentation elements that are part of the tree.*
- *$C \subseteq \{[v, b_{p'}] \mid p, p' \in P, p \neq p', v \in V_p \cup PV_p\}$ the set of connections in the tree.*

For a valid fragment $f = (P, C)$ the following conditions must hold:

1. *If $c_1, c_2 \in C$, $c_1 = [v_1, b_p], c_2 = [v_2, b_p], p \in P$ then $v_1 = v_2$, i.e., each binding point can be bound to only one variable.*
2. *If $c_1, c_2 \in C$, $p, p' \in P$ and $c_1 = [v, b_p], c_2 = [v, b_{p'}]$ then $p = p'$, i.e., each variable can be bound to only one binding point.*

3. $Unbound_f = \{p \in P | \neg\exists v \in \bigcup_{p' \in P} V_{p'} : [v, b_p] \in C\}$ *and* $|Unbound_f| = 1$, $root_f = p \in Unbound_f$
There is exactly one presentation element $p \in P$ *of the fragment* f *that is not bound to any other presentation element. This unbound presentation element is called the root element, denoted* $root_f$, *of the fragment and has the binding point* b_{root_f} *that forms the "entry point" of the fragment.*
4. *There is no sequence of connections* $c_1, \ldots, c_n$, *such that* $c_i = [v_i, b_{p_i}], i = 1 \ldots n-1$, *with* $v_{i+1} \in V_{p_i}$, *and* $v_1 \in V_{p_n}$. *This means that* f *is acyclic.*

Fragments form the building blocks of a multimedia document. They are the units that can be reused and recomposed in different multimedia documents. Therefore, the definition of a *complex media object* is needed. A complex media object *cm* encapsulates a fragment $f = (P, C)$ so that a fragment can simply be reused like a presentation element in any other fragment. A complex media element *cm* can be characterized as follows:

Definition 12.3.5 (Complex media element)
A complex media element $cm : [t_{cm}, b_{cm}, V_{cm}, PV_{cm}, f]$ *is a presentation element with* $t_{cm} = Complex \in T$, $f = (P, C)$ *denoting the fragment encapsulated by cm,* $b_{cm} = b_{root_f}$, $V_{cm} = \{v \in \bigcup_{p \in P} V_p \mid \forall q \in P : [v, b_q] \notin C\}$, *and* $PV_{cm} = \{pv \in \bigcup_{p \in P} PV_p \mid \forall q \in P : [pv, b_q] \notin C\}$.

The binding point of the root of the encapsulated fragment f is also the binding point of the complex media object *cm*. All variables and all projector variables in the fragment f that are not bound are *exported* and form the unbound variables and projector variables of the complex media object. For an illustration see Figure 12.4.

As complex media objects encapsulate fragments of multimedia documents, they offer a means of abstraction. Complex media objects can be treated like presentation elements and can be used in any other fragment, arbitrary complex. The export of unbound variables also allows for a later specialization of complex media objects. Hence, by complex media elements document templates can be encapsulated.

To encapsulate fragments that are specified in an external format we define, *external media elements*. An external media element *em* is also a complex media element. It encapsulates, however, not a fragment specified in ZyX , but the specification of an external fragment available in another data model. Like the complex media element, the external media element has assigned a set of variables V_{em}, projector variables PV_{em}, and one binding point b_{em}. However, the meaning of the variables and projector variables depends on the external document format.

With the definitions given so far it is possible to arrange presentation elements in certain relationships by means of connections. Though the connections of variables to binding points bring presentation elements in a relationship, the

semantics of this relationship is not yet defined. Therefore, our data model offers different types of operator elements which relate presentation elements with a certain semantics.

In the following, we present the element definitions of several groups of *temporal operators*, *projectors*, *selectors*, *interaction elements*, and *adaptation elements*. These elements determine the semantics that have to be interpreted by a presentation environment and mapped into the spatial, temporal, structural, interaction, and adaptive domain of a multimedia presentation. In the following definitions, only the domains of those tuple elements that characterize the element specific semantics are explicitly given.

12.3.3.2 Temporal Operator Elements. The *temporal operator elements* determine the temporal relationships between the presentation elements. As outlined above, our temporal model is based on Interval Expressions [5]. In the following, we present the definition, specific parameters, and semantics of the temporal operator elements *par*, *seq*, *loop*, and *delay*. For an illustration of the temporal operator elements see Figure 12.8.

The semantics of the *par operator element* is that the presentation elements bound to its variables are to be presented in parallel by a presentation engine. The element is defined as follows:

Definition 12.3.6 (Temporal operator element — *par*)
The temporal operator element par : $[t_{par}, b_{par}, V_{par}, PV_{par}, finish, lipsync]$ *is a presentation element with* $t_{par} = Par \in OT$, $V_{par} = \{v_1, \ldots, v_n\} \subseteq VAR$, $finish \in \{1, \ldots, n, min, max\}$, *and* $lipsync \in \mathcal{N}_l$.

The *par* operator element offers two parameters to control the synchronization of parallel presentation. The parameter $finish$ determines which one of the n presentation elements terminates the parallel presentation, i.e., the one with the minimal presentation time by setting $finish = min$, the maximum presentation time by setting $finish = max$, or a dedicated presentation element bound to v_i, by setting $finish = i, i \in \{1, \ldots, n\}$. If the parameter $lipsync = 0$ no lip synchronization is specified. If the value of $lipsync = i$, $i > 0$, the presentation of the presentation elements bound to $v_1, \ldots, v_n$ is carried out in lip synchronization and the presentation element bound to v_i forms the master of the synchronization.

The semantics of the *seq operator element* is that a presentation engine presents the presentation elements that are bound to it in sequence. The presentation of a *seq* operator element starts the sequential presentation of the presentation elements that are bound to the variables $v_i, i = 1 \ldots n$ in the order of $v_1, v_2, \ldots, v_n$. The presentation of the *seq* operator element ends with the end of the presentation of the element bound to v_n. The *seq* operator element is defined as:

Definition 12.3.7 (Temporal operator element — *seq*)
The temporal operator element seq : $[t_{seq}, b_{seq}, V_{seq}, PV_{seq}]$ *is a presentation element with* $t_{seq} = Seq \in OT$, *and* $V_{seq} = \{v_1, \ldots, v_n\} \subseteq VAR$.

The semantics of a *loop operator element* is that its presentation starts the repeated presentation of the single presentation element bound to $v \in V_{loop}$. The presentation is repeated r times and stops after the r^{th} presentation of the presentation element.

Definition 12.3.8 (Temporal operator element — *loop*)
The temporal operator element loop : $[t_{loop}, b_{loop}, V_{loop}, PV_{loop}, r]$ *is a presentation element with* $t_{loop} = Loop \in OT$, $|V_{loop}| = 1$, *and* $r \in \mathcal{N}$.

The *delay* operator element models a temporal delay of t milliseconds. It can be seen as an "empty" media element that is presented for t milliseconds.

Definition 12.3.9 (Temporal operator element — *delay*)
The temporal operator element delay : $[t_{delay}, b_{delay}, V_{delay}, PV_{delay}, t]$ *is a presentation element with* $t_{delay} = Delay \in OT$, $V_{delay} = \emptyset = PV_{delay}$, *and* $t \in \mathcal{N}$.

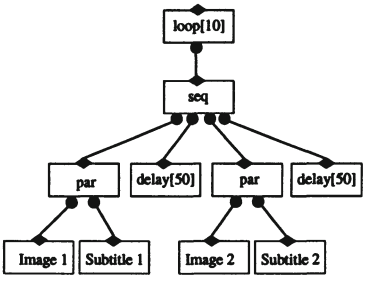

Figure 12.8: Fragment illustrating the usage and semantics of the temporal operator elements

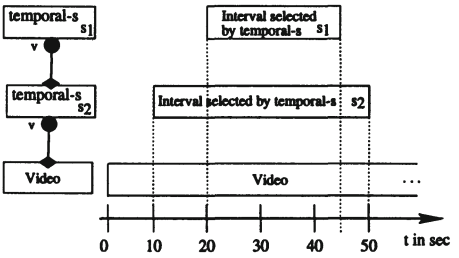

Figure 12.9: Sample fragment illustrating the usage and semantics of the temporal selector element *temporal-s*

12.3.3.3 Projectors. To add "layout" information to a presentation element, it can have 0 to n projector variables pv that can be used to bind *projector elements* to the presentation element. The projector elements can be statically bound to projector variables at authoring time or just before the presentation of a document. Projector elements are also presentation elements that determine *how* presentation elements are presented. A projector element can not bind other presentation elements and, therefore, for all projector elements $V_{proj} = \emptyset$ and $PV_{proj} = \emptyset$. The model offers four different projector elements, *spatial-p*, *temporal-p*, *acoustic-p*, and *typographic-p*, some of which we define in the following.

First, an auxiliary definition of the notion of a *successor* in a fragment needed for subsequent definitions is given:

Definition 12.3.10 (Successor)
Let $\mathcal{F}$ *denote the set of all fragments. We then define a function expand* : $\mathcal{F} \rightarrow$

$\mathcal{F}$ that computes for a fragment f the fragment that is semantically equivalent to f but does not contain any complex media element. expand(f) recursively replaces each complex media element in f by the fragment that the complex media element encapsulates.

Be $f \in \mathcal{F}$ a fragment, expand(f) $= (P, C)$, and $p, p' \in P$ presentation elements. Then the following direct and indirect successor relationships hold:

1. *p' is direct successor of $p \iff \exists [v, b_{p'}] \in C : v \in V_p$.*
2. *p' is indirect successor of $p \iff p'$ is not a direct successor of p and there exists a sequence $succ_1, \ldots, succ_n, n \in \mathcal{N}$ with $succ_1$ is direct successor of p, $succ_i$ is direct successor of $succ_{i-1}, i = 2, \ldots, n$, and p' is direct successor of $succ_n$.*
3. *p' is successor of $p \iff p'$ is direct or indirect successor of p.*

For example, in Figure 12.4 the video media element and the parallel element are direct successors of the root sequential element. The audio element is an indirect successor of the root sequential element and a direct successor of the parallel element. There is no successor relationship between video and the audio media element.

The presentation semantics of the *spatial projector element spatial-p* (Definition 12.3.11) bound to a presentation element p is that the presentation engines "projects" the visual presentation of p on a rectangular presentation area, which is defined by the projector element. The parameters x and y define the position of the upper left corner of a rectangle with the given *width* and *height*. The parameter *priority* defines the order of the overlapping of visual objects so that an object with a higher priority value covers objects with a lower priority value. The parameter *unit* defines the measurement unit to specify whether the values $x, y, width, height$ are given in pixel or in percent of a presentation window.

Definition 12.3.11 (Spatial projector element — *spatial-p*)
The spatial projector element spatial-p : $[t_{spatial-p}, b_{spatial-p}, V_{spatial-p}, PV_{spatial-p}, x, y, width, height, priority, unit]$ is a presentation element with $t_{spatial-p} =$ Spatial-P $\in OT$, $V_{spatial-p} = PV_{spatial-p} = \emptyset$, x, y, priority $\in \mathcal{N}_l$, width, height $\in \mathcal{N}$, and unit $\in \{pixel, percent\}$.

A spatial projector applies not only to the presentation element it is bound to but to all successors of this presentation element. If the projector is bound to a media object then the media object is scaled to the presentation area defined by the projector's parameters. If the *spatial-p* element of a presentation element p defines a presentation rectangle, the spatial coordinates and extensions of the successors of p are seen in the context of that rectangle and not of the entire presentation window.

The presentation semantics of the *temporal projector element temporal-p* (Definition 12.3.12) bound to a presentation element p is that a presentation engine presents the element p with the given playback direction and speed. The parameter *direction* specifies, whether the presentation element (and its

subtree) is presented in forward (1) or in backward direction (−1). The actual playback speed is computed by multiplying the original playback speed with the factor given by the speed parameter.

Definition 12.3.12 (Temporal projector element — *temporal-p*)
The temporal projector element temporal-p : $[t_{temporal-p}, b_{temporal-p}, V_{temporal-p}, PV_{temporal-p}, direction, speed]$ *is a presentation element with* $t_{temporal-p} =$ *Temporal-P* $\in OT$, $V_{temporal-p} = PV_{temporal-p} = \emptyset$, *direction* $\in \{-1, 1\}$, *and speed* $\in \Re^+$.

Like the spatial projector element a temporal projector element applies not only to the presentation element p it is bound to but to all successors of that presentation element. If, for example, the *temporal-p* projector of a presentation element p defines *speed* $= 2$ and a successor p' of p has a temporal projector that also defines *speed* $= 2$ then in fact the successor p' is presented at a speed factor of 4.

In the same way an acoustic projector element and a typographic projector element are defined. The acoustic projector element *acoustic-p* affects the, e.g., volume, balance, base, and treble of the presentation of a presentation element p, while the typographic projector element *typographic-p* affects parameters like the font, size, and style of the presentation of p.

12.3.3.4 Selectors. The model offers *selector elements* to reuse parts of media elements and fragments, i.e., spatial regions, temporal intervals. A *temporal selector element temporal-s* (Definition 12.3.13) is a presentation element that can bind one other presentation element p. The presentation semantics of this element is that the presentation of the direct and indirect successors of p is started *start* milliseconds after the original starting point of the fragment and lasts for *duration* milliseconds.

Definition 12.3.13 (Temporal selector element — *temporal-s*)
The temporal selector element temporal-s: $[t_{temporal-s}, b_{temporal-s}, V_{temporal-s}, PV_{temporal-s}, start, duration]$ *is presentation element with* $|V_{temporal-s}| = 1$, $t_{temporal-s} =$ *Temporal-S* $\in OT$, *and* $start, duration \in \mathcal{N}_t$.

A *spatial selector spatial-s* (Definition 12.3.14) element can bind one other presentation element p, which can be a visual media element like an image or a video but also a complex media element. The spatial selector selects a spatial area from p. The presentation semantics is that the presentation engine presents only those visual parts of p and its successors that are visible in the rectangular area that is specified with the element's parameters $x, y, width$, and $height$.

Definition 12.3.14 (Spatial selector element — *spatial-s*)
The spatial selector element spatial-s: $[t_{spatial-s}, b_{spatial-s}, V_{spatial-s}, PV_{spatial-s}, x, y, width, height]$ *is a presentation element with* $t_{spatial-s} =$ *Spatial-S* $\in OT$, $|V_{spatial-s}| = 1$, x, $y \in \mathcal{N}_t$, *and* $width, height \in \mathcal{N}$.

The application of selector elements is context sensitive. That is, it applies to the entire subtree of the presentation element bound to it. Selector elements can be organized in a hierarchy. Then, each selector element is applied relatively to the context of the subtree bound to it. Consider for example two temporal selector elements s_1 and s_2, $s_1 = [\textit{Temporal-S}, b_{s_1}, \{v_{s_1}\}, \emptyset, 10000, 25000]$ and $s_2 = [\textit{Temporal-S}, b_{s_2}, \{v_{s_2}\}, \emptyset, 10000, 40000]$. Let s_2 be a direct or indirect successor of s_1 then the selected temporal interval defined by s_1 is defined relative to the temporal interval specified by s_2 (see Figure 12.9).

12.3.3.5 Interaction Elements. To support the requirement of interactive multimedia presentations, the model offers different *interaction elements.*

The *link* interaction element (Definition 12.3.15) can be used to model navigational interactions between multimedia documents. Herewith hypertext structures can be modeled.

Definition 12.3.15 (Interaction element — *link*)
The interaction element link : $[t_{link}, b_{link}, V_{link}, PV_{link}]$ *is a presentation element with* $t_{link} = Link \in OT$, $V_{link} = \{v_1, \ldots, v_n, t_1, \ldots, t_n\}$, *and* $n \in \mathcal{N}$.

The *link* interaction element defines a set of links between the presentation elements bound to $v_i \in V_{link}, i = 1 \ldots n$, and the presentation elements bound to $t_i \in V_{link}$. Each presentation element bound to v_i represents the anchors of a link while the presentation element bound to the variable t_i specificies the target of the link. The presentation semantics of the *link* element is that a user interaction, e.g., a mouse click, with a presentation element bound to v_i, e.g., an image, starts the presentation of the target presentation element bound to t_i, e.g., a slide show. The presentation of the link element terminates when an interaction with one of the anchor elements occurred or the presentation of all anchor elements is terminated.

While a *link* interaction element is intended for the navigation *between* documents, the *menu* interaction element (Definition 12.3.16) is provided to allow for navigation *within* a document, i.e., the selection of one out of a set of presentation paths.

Definition 12.3.16 (Interaction element — *menu*)
The interaction element menu : $[t_{menu}, b_{menu}, V_{menu}, PV_{menu}, mode]$ *is a presentation element with* $t_{menu} = Menu \in OT$, $mode \in \{vanish, prevail\}$, $V_{menu} = \{v_1, \ldots, v_n, t_1, \ldots, t_n\}$, *and* $n \in \mathcal{N}$.

Similar to the *link* interaction element, the *menu* interaction element defines a set of selectable presentation elements bound to $v_i \in V_{menu}, i = 1 \ldots n$. The presentation elements bound to $t_i \in V_{menu}, i = 1 \ldots n$ represent the corresponding target elements of the selection. The presentation semantics of the *menu* element is that on presentation of the *menu* element, the engine starts in parallel the presentation of the elements bound to $v_i \in V_{menu}, i = 1 \ldots n$. On selection of a presentation element bound to v_i, the engine presents the target element of the selection bound to t_i. That is, a selection of the element

bound to v_i corresponds to the presentation of the target bound to t_i. If parameter *mode* = *vanish*, the engine finishes the presentation of all presentation elements bound to $v_j, j = 1 \ldots n$, and starts the presentation of the presentation element bound to t_i. If parameter *mode* = *prevail*, the engine "merges" the presentation of the presentation element bound to t_i with the currently running presentation. If no element is selected, the presentation of the *menu* element stops as soon as the presentation of all presentation elements bound to $v_i, i = 1 \ldots n$, is finished.

We have also defined two further types of interaction elements, *interactive projector elements* and *interactive selector elements*. These elements comply in general with the projector and selector elements presented before, but they have an additional "interactive" component. For each projector element and selector element, a corresponding interaction element is offered. With these interactive elements design interactions of multimedia presentations can be modeled.

For example, an interactive projector element *temporal-pi* is an interactive variant of the *temporal-p* projector element. Its presentation semantics is that, in addition to the specified temporal projection, the presentation engine offers a user to interactively adjust the parameters *direction* and *speed*.

12.3.3.6 Adaptation elements. Our model offers the two elements *switch* and *query* which allow for the adaptation of a multimedia presentation according to the user's interest and system environment that are described in a *global profile GP* by means of attribute value pairs. The *switch* adaptation element (Definition 12.3.17) serves the purpose to specify different presentation alternatives with regard to *GP*.

Definition 12.3.17 (Adaptation element — *switch*)
The adaptation element switch : $[t_{switch}, b_{switch}, V_{switch}, PV_{switch}, M_1, \ldots, M_n]$ *is a presentation element with* $t_{switch} = Switch \in OT$, M_i *denoting sets of attribute-value pairs,* $V_{switch} = \{v_1, \ldots, v_n, v_{default}\}$, *and* $n \in \mathcal{N}$.

The semantics of the *switch* element is that upon its presentation the presentation engine sequentially evaluates the metadata given with the *GP* against the sets of metadata $M_i, i = 1 \ldots n$. Let $M_j, j \in \{1, \ldots, n\}$ be the set of metadata which matches best *GP*. Then, the fragment bound to v_j is presented. If there is no suitable set of metadata among $M_1, \ldots, M_n$, the presentation element bound to $v_{default}$ is selected for presentation. The presentation of the *switch* element terminates when the presentation of the selected presentation element is finished.

In cases in which the presentation alternatives of a document are not known at authoring time, the *query* element (Definition 12.3.18) is provided. The *query* element is a placeholder for a fragment. It specifies a "query" which selects a fragment at presentation time from all available fragments. Therefore, we enhance the definition of a fragment such that it includes metadata, i.e., $f = (P, C, M)$ with M being a set of attribute-value pairs. This metadata describes both the content of a fragment f and technical features of the fragment like the network bandwidth needed for its presentation.

Definition 12.3.18 (Adaptation element — *query*)
The adaptation element query : $[t_{query}, b_{query}, V_{query}, PV_{query}, M]$ *is a presentation element with* $t_{query} = Query \in OT$, M *denoting a set of attribute-value pairs, and* $V_{query} = \emptyset$.

The semantics of the *query* element is that the presentation engine evaluates the metadata specified with $M \cup GP$ against the metadata given with all fragments known to the system. Then the fragment with the best match with respect to M and the profile GP is selected for presentation. This allows to dynamically select the most suitable fragment at presentation time taking into account the actual user interest and system environment. The presentation of the *query* element terminates when the presentation of the selected fragment is finished.

12.4 CONCLUSION AND FUTURE WORK

Starting out with the requirements of the Cardio-OP project, which calls for the support of *reusability*, *interaction*, *adaptation*, and *presentation-neutral description* of the structure and content of multimedia documents, we sketched our analysis of existing relevant multimedia document models. As these models do not meet the project's requirements, we introduced our new ZyX model that gives the necessary support. We outlined the design considerations and the basic concepts followed by a formal framework of the ZyX primitives.

The ZyX model has been implemented as a DataBlade module for the object-relational database system Informix Dynamic Server/Universal Data Option under Sun Solaris, following the architectural framework initially presented in [12, 3]. Ongoing work includes the identification and realization of possible optimizations of the implementation of the DataBlade.

The formal description served as the basis for the definition of an XML DTD for the ZyX model[2]. This will enable access to content stored in the Cardio-OP repository by future XML-capable browsers and we can also think about storing ZyX documents in an SGML/XML-capable database system in the future, following the approach taken in [2].

Furthermore, we are working on a generic presentation engine for ZyX documents which includes support for continuous MPEG video streams based on an extension of the L/MRP buffer management technique [14].

Further work is needed to extend the representation of meta data, its usage for querying the Cardio-OP repository taking into account the various approaches discussed in, e.g., [16], and to exploit appropriate indexing techniques.

[2]The element definitions for a very first version of an XML DTD is available under URL www.informatik.uni-ulm.de/dbis/Cardio-OP/cardioopxml.dtd

Acknowledgments

We would like to thank Utz Westermann for his contributions to the design and implementation of the ZyX model and to preparing the final version. We would also like to thank Christian Heinlein for his valuable comments on the paper.

References

[1] Allen, J. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843.

[2] Böhm, K., Aberer, K., and Klas, W. (1997). Building a Hybrid Database Application for Structured Documents. In *Multimedia - Tools and Applications, Accepted for Publication*, Dordrecht. Kluwer Academic Publishers.

[3] Boll, S., Klas, W., and Löhr, M. (1996). Integrated Database Services for Multimedia Presentations. In Chung, S., editor, *Multimedia Information Storage and Management*. Kluwer Academic Publishers, Dordrecht.

[4] Bray, T., Paoli, J., and Sperberg-McQueen, C. (1998). *Extensible Markup Language (XML) 1.0 – W3C Recommendation 10-February-1998*. W3C, URL: http://www.w3.org/TR/1998/REC-xml-19980210.

[5] Duda, A. and Keramane, C. (1995). Structured temporal composition of multimedia data. In *Proc. IEEE International Workshop on Multimedia-Database-Management Systems*, Blue Mountain Lake.

[6] Hoschka, P., Bugaj, S., Bulterman, D., et al. (1998). *Synchronized Multimedia Integration Language – W3C Working Draft 2-February-98*. W3C, URL: http://www.w3.org/TR/1998/WD-smil-0202.

[7] ISO/IEC (1986). *Information processing - Text and Office Systems - Standard Generalized Markup Language (SGML)*. ISO/IEC IS.

[8] ISO/IEC (1992). *Information Technology - Hypermedia/Time-based Structuring Language (HyTime)*. ISO/IEC IS.

[9] ISO/IEC (1995). *Information Technology – Coding of Multimedia and Hypermedia Information – Part 5: Support for Base-Level Interactive Applications, ISO/IEC IS 13522-5*. ISO/IEC IS.

[10] ISO/IEC (1996). *Information Technology – Coding of Multimedia and Hypermedia Information – Part 6: Support for Enhanced Interactive Applications, ISO/IEC IS 13522-6*. ISO/IEC IS.

[11] ISO/IEC (1997). *Information technology – Coding of multimedia and hypermedia information – Part 1: MHEG object representation ISO/IEC 13522-1*. ISO/IEC IS.

[12] Klas, W. and Aberer, K. (1997). Multimedia and its Impact on Database System Architectures. In Apers, P., Blanken, H., and Houtsma, M., editors, *Multimedia Databases in Perspective*. Springer, London.

[13] Little, T. D. C. and Ghafoor, A. (1993). Interval-based conceptual models for time-dependent multimedia data. *IEEE Transactions on Knowledge and Data Engineering*, 5(4).

[14] Moser, F., Kraiß, A., and Klas, W. (1995). L/MRP: A Buffer Management Strategy for Interactive Continuous Data Flows in a Multimedia DBMS. In *Proceedings VLDB 1995*, USA. Morgan Kaufmann.

[15] Newcomb, S., Kipp, N., and Newcomb, V. (1991). HyTime – The Hypermedia/ Time-Based Document Structuring Language. *Communications of the ACM*, 34(11).

[16] Sheth, A. and Klas, W. (1998). *Multimedia Data Management - Using Metadata to Integrate and Apply Digital Media.* McGraw-Hill, New York.