Cardio-OP

Anatomy of a multimedia repository for cardiac surgery

Susanne Boll, Christian Heinlein(*), Utz Westermann

Multimedia Information Systems
Institute for Computer Science and Business Informatics
Vienna University, Austria
{Susanne.Boll,Gerd-Utz.Westermann}@univie.ac.at

(*)Computer Structures
Faculty of Computer Science
University of Ulm, Germany
{heinlein}@informatik.uni-ulm.de

Abstract

Multimedia applications call for a comprehensive support for the entire cycle from the production to the consumption of multimedia content by a multimedia information system. The solutions we find here are insular and concentrate on the single aspects of storage and maintenance of multimedia information, multimedia retrieval, authoring, and delivery of the multimedia content. However, a continuous, integrated solution is lacking.

To support the entire multimedia producer-consumer cycle, we have been developing the concept of a database-driven multimedia repository. This multimedia repository provides sophisticated multimedia storage, maintenance, retrieval, authoring, and presentation support to the different users of the system. We validated and proved the applicability of our approach in the research project "Gallery of Cardiac Surgery" (Cardio-OP) in which we developed a multimedia training and education system in the domain of cardiac surgery. In this project context, the multimedia repository and the tools developed showed that they can support next generations of online multimedia training and education systems by a much more intuitive and economical maintenance, composition, and delivery of multimedia teaching and training material to the users.

Keywords

Multimedia database systems, multimedia repositories, medical multimedia applications, multimedia training

I. Introduction

This work evolves from the need for the comprehensive, sophisticated support of the multimedia consumer-producer cycle, i.e., to develop a multimedia information system that is able to create, maintain, retrieve, compose, and deliver multimedia informations to the clients of the system. In the past, much of the educational and training material has been provided by means of print media. In recent years more and more *multi*media has been "discovered" as a proper means to impart knowledge, and electronic media like CD-ROM have been employed for the distribution of this educational material. However, this was and is from our view a dead-end development.

Multimedia productions are very expensive due to several aspects. One needs costly technical equipment and specialists for its operation to prepare the raw media data. Here, high costs also arise from qualified personnel needed for planning and carrying out the media acquisition. For the composition of the raw material into multimedia productions additional costs arise. In the educational sector, for the preparation of sophisticated multimedia content one not only needs experts which have the expertise of the application domain but also experts in commercial multimedia authoring tools and their programming languages. The resulting, dearly purchased multimedia content mostly forms pre-orchestrated proprietary multimedia productions targeted at a specific user group - small changes to the topic or targeted user group triggers the expensive production process again.

However, many application domains demand a much better comprehensive support than that for a flexible management, composition, and delivery of their multimedia material for different user groups over different distribution channels. This is due to efficiency and economical reasons. The high effort and the unsatisfying support still withholds many typical content provides like small and medium sized publishing houses to enter a new market and participate in the production and distribution of multimedia content. The way we see to reduce the effort and invite multimedia content producers to create and deliver multimedia material to their customers is by a better exploitation of the content. The acquisition of high-quality material will, from our point of view, still be a time consuming and costly task. However, a better exploitation of the available content puts the high production cost in a better perspective and yields a much better cost-value ratio. Therefore, for efficiency and economical reasons, multimedia content producers need support for the creation and distribution of flexible multimedia productions in which the material can be easily managed, retrieved and flexibly (re)composed into multimedia presentations in many different contexts or different user groups. This is where new technologies like those presented in this paper fall on fertile ground.

We have been designing and developing a multimedia repository that provides support for the entire producer-consumer cycle to the system's users. Our research project "Cardio-OP" served as an application to validate the concepts of the approach. We illustrate the features of the multimedia repository and tools before the project background.

The remainder of the paper is organized as follows: Section II introduces the Cardio-OP project and its specific goals. Section III illustrates the multimedia producer-consumer cycle, motivates the employment of a multimedia repository, and presents the system's basic architecture. Section IV presents in detail the system components of the repository and the integrated tools that jointly make up the multimedia information system. We conclude the paper in Section V and give an outlook to our ongoing research activities.

II. The Cardio-OP project

A. Project Background and Motivation

Cardiac surgery is a field in medicine dealing with the operative treatment of congenital and acquired diseases of the heart and its great vessels. This discipline is facing a rapid development of new operative techniques and new technologies. The complexity of the subject requires continuous training, education, and information for different groups in heart surgery: physicians in the field of cardiac surgery, medical students, perfusionists, and patients. Books usually contain sketches or pictures about the operative invasion which can not really provide an illustrative training for the actual operation techniques, especially, as the subject changes so fast that the low frequency of new editions cannot cope with these. There is evidence in the literature that efficacy and efficiency of education and training in different fields of medicine may be improved by the use of multimedia information systems.

The production of illustrative multimedia teaching units, however, is very expensive as both specific technical equipment is needed, e.g., to take videos during a surgery, and the post-processing of the material needs expertise in multimedia editing *and* cardiac surgery. The available multimedia content so far is mainly delivered on CD-ROM or distributed over the web, targeted at one specific user group in one specific instructional context. The produced computer-based learning programs often show deficiencies in the support of individual learning styles and in providing information corresponding to the individual learner's needs.

Next generations of online multimedia training and education applications call for new approaches for the creation, storage, maintenance, commercial marketing, and publishing of multimedia content. In our research project "Gallery of Cardiac Surgery" [1] a multimedia information system is envisioned that will serve as a common information and education base for its different types of users in the domain of cardiac surgery. A broad range of users, from physicians, students, medical lecturers to patients, are provided with multimedia information according to their user specific request to the multimedia information system, their background knowledge, their different understanding of the selected subject, their location and technical infrastructure. The users will have access to instructional applications like a "multimedia book" on operative techniques, a module for training of decision making for operations, a module to create and present lectures, and a module for patient information.

The information system is intended to support the entire production process of this multimedia content: For an economical usage of the multimedia content the material, stored and managed in a central repository, is to be reused in many different application contexts. The multimedia information can be selected and composed according to the users' needs and interests to provide them with a personalized view of the material. The resulting multimedia productions can be delivered over different output channels to the users' heterogeneous system environments. The content is adapted to available service quality at the targeted presentation environment without explicitly producing a different version for each of the different environments.

The technology developed is intended to be applicable to other application domains as well, such as continuous education and training programs for employees in production processes.

Both, the demand for innovative multimedia training and education systems in the fast developing field of Cardiac Surgery and the need for new means of economically creating, maintaining, marketing, and distributing various kinds of professional knowledge in terms of multimedia content by publishing companies constitute the main driving force behind the Cardio-OP project.

For an economical usage of the multimedia content the material, stored and managed in some kind of a (central) repository, is to be reused in many different application contexts. The multimedia information must be selected and composed according to the users' needs and interests to provide them with a personalized view of the material. It is desirable that one multimedia production can be delivered over different output channels to the users' heterogeneous system environments. The content should be adapted to available service quality at the targeted presentation environment without explicitly producing a different version for each of the different environments.

B. The project partners

The application requirements were provided by the University Hospital of Ulm (Cardiological and Surgical Clinics), the University Hospital of Heidelberg (Surgical Clinic) and an associated rehabilitation hospital. Further project partners were the FAW (Research Institute for Applied Knowledge Processing), Ulm, the publishers Barth-Verlag and dpunkt-Verlag, Heidelberg, as well as the ENTEC GmbH, St. Augustin.

III. The Cardio-OP multimedia repository

In this section, we motivate and introduce the notion of a multimedia repository as a multimedia information system adequately supporting the complete chain of working tasks occurring in the process of multimedia production, ranging from media acquisition, value-adding, and authoring to the delivery and presentation of multimedia content. We first illustrate the different steps that are involved on the long way from multimedia production to consumption (III-A). We then give our definition of the term "multimedia repository" (III-B) before concluding this section with an architectural overview of the implementation of a multimedia repository that we provided during the Cardio-OP project (III-C).

A. The multimedia producer-consumer cycle

From a user's perspective, the Cardio-OP multimedia information system needs to adequately support the different working tasks on the way from the acquisition of multimedia content to its final consumption. Hence, medical teachers, students, and patients must be provided with a tool set to create sophisticated multimedia content, to deliver, and to consume the multimedia content over a heterogeneous network and system infrastructure. Arranged in a *multimedia producer-consumer cycle*, Figure 1 illustrates the different working tasks and the required functionalities: acquiring, value-adding, browsing and retrieving, authoring, delivering, and presenting multimedia information. As a proper basis for this variety of tasks, efficient *storage and management* of the generally expensively produced multimedia content at a fine granularity are indispensable. Thus, the different tasks of the multimedia producer-consumer cycle are arranged circularly around the central multimedia storage and management.

Multimedia production starts out from the phase of *media acquisition* in which the different media types are digitally created and encoded in appropriate media formats, e.g., by taking digital video, scanning images, or keying in text. As media acquisition does not necessarily make the multimedia content available in its final form, *value-adding* postprocessing might be necessary. For example, a digital video can be processed by automatic scene detection software. Domain experts might further annotate the media object with semantic, content-based metadata to provide additional semantics enabling effective, content-based retrieval of the video. For the authoring process, i.e., the composition of continuous and discrete media elements into a logically coherent multimedia presentation, the *browsing* in the media pool and the targeted *retrieval* of media elements for this composition process are necessary prerequisites. *Multimedia authoring* means the composition of media into multimedia documents, i.e., temporally and spatially interrelating single media items and specifying possible interactions between the consumer and the presentation of the document. With the authoring of documents, the set of types of multimedia information is complete: media elements, metadata, and multimedia documents. As multimedia documents also include continuous media of potentially high data volume, an adequate means for *delivery* of this media data must be provided. The users neither want to wait for a long time for a requested information nor do they accept a fluttering play back of a continuous stream. The support for delivering multimedia hence must include continuous streaming of multimedia to the consumers and must be integral part of the presiding multimedia presentation process. The output devices of such presentations nowadays are not limited to the home PC; we rather face a variety of different network connections and output devices here. The delivery and presentation of multimedia content should be automatically adapted to the respective output devices' requirements. The *multimedia presentation* of a multimedia document subsumes the rendering of the participating media elements, the realization of the temporal and spatial layout, and the user interaction. The rendering of multimedia content, tailored to the different output channels, for the consumers completes the multimedia producer-consumer cycle.

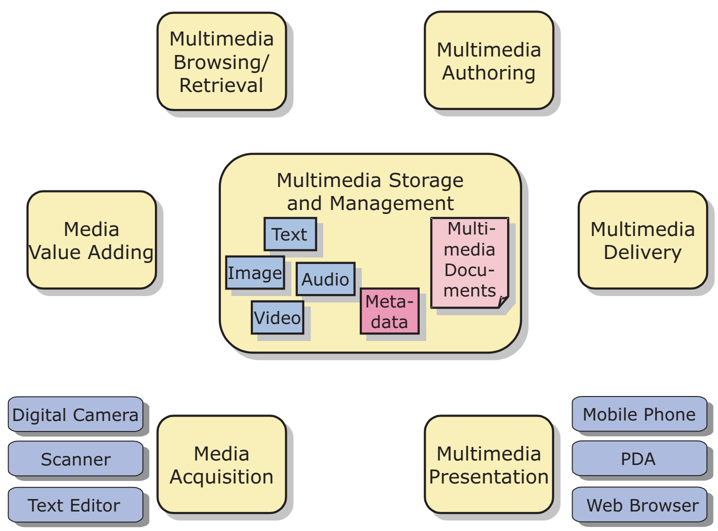

Fig. 1. The multimedia producer-consumer cycle

B. Multimedia repositories

We decided to support the multimedia producer-consumer cycle by means of a *multimedia repository*. The notion of a multimedia repository, however, is often used to denote just a multimedia database hosting a collection of media objects. In contrast to this, we were concerned with the integration of presentation services with a multimedia database management system in earlier research [2], [3]. The central idea was to offer *presentation independence* as a multimedia database service just like a traditional DBMS offers services like data independence and multiuser support [4]. In the Cardio-OP project, consequently continuing this integrative approach, not only the presentation, but also the production of multimedia data must be well integrated with the underlying media management system, e.g., a database system.

Hence, we define a multimedia repository to be *an information system suitably supporting the entire multimedia producer-consumer cycle*: it must provide both sophisticated system components for the import, value-adding, storage, management, retrieval, authoring, delivery, and presentation of multimedia information and, well integrated with these components, graphical tools that give the users of the system an adequate access to and visualization of the system's functionality.

According to this definition, our group has developed the DBMS-driven Cardio-OP multimedia repository. It stores the multimedia content annotated with metadata from the application domain, supports its content-based retrieval, and integrates the authoring, delivery, and presentation of multimedia content. As the information system is common to different user groups, it is designed to support flexible, fine-grained reuse of the multimedia material in different contexts. Though the multimedia repository has been designed with the specific requirements of Cardio-OP's application domain in mind, we strived to provide, to an extent as great as possible, a domain-independent solution. Hence, most parts of our multimedia repository are generic and can be directly employed in other application domains as well.

C. The architectural approach

Starting out from our considerations concering the multimedia producer-consumer cycle, we developed an architecture of a multimedia repository supporting all phases of this cycle. Figure 2 depicts this architecture consisting of three main building blocks: a database-driven *Media Server*, a *Continuous Media Streaming Facility*, and a *Tool Suite*.

The Media Server handles all the media data, metadata, and multimedia documents occurring in the Cardio-OP multimedia repository. It is founded on an object-relational database management system since the advantages of employing database management

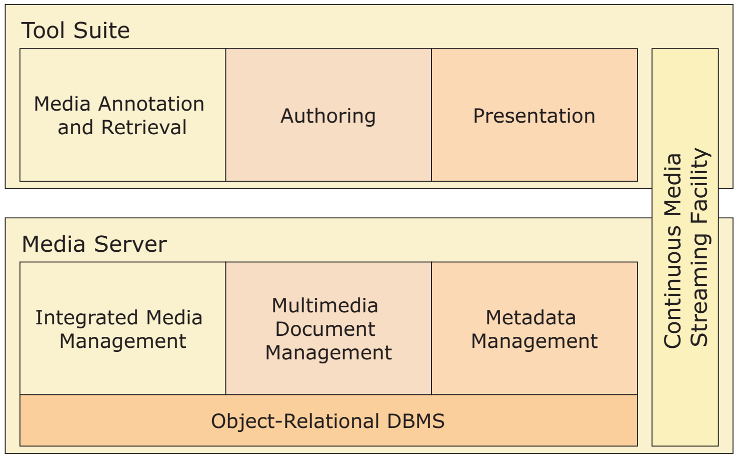

Fig. 2. Architecture of the Cardio-OP multimedia repository

systems (DBMS) with multimedia data have long been recognized [2], [3], [5], [6]. A DBMS readily provides support for several fundamental requirements of a multimedia information system as envisioned in the Cardio-OP project. Among these are multiuser access, support of transactions, query languages, and stable recovery mechanisms. The latter is especially noteworthy considering the high value of media content and descriptional metadata arising from the high costs of media production. An object-relational DBMS (ORDBMS) combines these strengths with profound extensibility necessary to seamlessly integrate multimedia data of high volume and complex structure as well as basic multimedia services for enabling content-based retrieval and presentation of continuous media data and multimedia documents.

The architecture of the Media Server comprises an ORDBMS which is extended with facilities for *integrated media management*, *multimedia document management*, and *management of metadata*. The facility for integrated media management offers a uniform view onto and uniform access to atomic media data of different type (video, audio, image, etc.) held at possibly different storage locations (file system, web server, media server, etc.). The multimedia document management facility extends the ORDBMS with capabilities for the fine-grained modeling of structured multimedia documents using the various atomic media data. Documents or parts of documents can be reused at arbitrary level of granularity. Finally, the metadata management extension integrates semantic description of both atomic media data and multimedia documents as well as content-based retrieval according to these descriptions with the ORDBMS. The semantic description of the content is based on domain-specific taxonomies.

Integral part of the repository is the Tool Suite offering a variety of tools which enable users to effectively work with the Media Server and to consume its content. To that end, tools for *media annotation and retrieval*, for *authoring*, and *presentation* of multimedia documents are provided. The tools for media annotation and retrieval offer easy-to-use interfaces to end-users for the import of atomic media data to the Media Server, for its annotation with describing concepts taken from the domain-specific taxonomies, and for the retrieval of the content using these annotations. The authoring tools facilitate the structured composition of multimedia documents at the domain level by domain experts. These authoring tools tightly integrate the tools for media retrieval and annotation to achieve a high degree of reuse of the content managed by the Media Server. Finally, the Tool Suite comprises a presentation engine enabling the synchronized presentation of the multimedia documents over a network.

In order to achieve smooth presentation of multimedia documents containing continuous media, the architecture of the multimedia repository provides a Continuous Media Streaming Facility [7]. It constitutes a bridge for the delivery of high-volumed continuous media data from the Media Server to the Tool Suite. More precisely, it enables the adaptive streaming of MPEG-1 video over a network taking into account potential user interactions that might occur during presentation.

IV. Components of the Cardio-OP multimedia repository

In the following, we illustrate the components of the Cardio-OP repository developed in the project following our architecture outlined before. We begin with the presentation of the Media Server and its components (IV-A) and continue with the Continuous Media Streaming Facility (IV-B) and the Tool Suite (IV-C). We conclude this section by reflecting on how the components of the Cardio-OP multimedia repository support the multimedia producer-consumer cycle, and to which extent these components constitute generic solutions.

A. The Media Server

As already motivated in the description of the architectural approach in Section III-C, we have chosen object-relational database management technology as the foundation of the Media Server to deal with the technical challenges arising from the need to seamlessly integrate support for multimedia data and services in a DBMS. In particular, we employ the Informix Dynamic Server 2000 (IDS 2000) [8] as this system is profoundly extensible by new data types, new native, user-defined routines, and new secondary access methods running in the DBMS kernel promising high efficiency of these extensions[1]. It already comes with some extensions concerning support for multimedia datatypes. During the project, we have extended the system by additional *DataBlade modules* enabling the integrated media management (IV-A.1), multimedia document management (IV-A.2), and metadata management (IV-A.3) to constitute an adequate Media Server in terms of the architecture of the Cardio-OP multimedia repository.

A.1 Integrated media management

The heart of the Cardio-OP multimedia repository is the sophisticated, integrated management of media data of *different* media types among which are video, audio, image, 3D animation, and text. There are several commercial DataBlade modules available extending the IDS 2000 with sophisticated support for different media types. It is highly desirable to employ such DataBlade modules as far as possible. When it comes to combine these DataBlade modules supporting different media types to achieve *multi*media management, however, certain important issues have to be considered:

- *Storage locations:* Each of the media type-specific commercial DataBlades support transparent access to media data at different storage locations such as file systems, binary large objects in databases, or web servers. However, the supported storage locations and the mechanisms to access these locations differ among the DataBlades. Hence, there is no transparent access to media data across the boundaries of the commercial DataBlades.
- *Technical metadata:* Each media type can be described by technical metadata, e.g., media size in bytes, framerate of videos, resolution of images, etc. The commercial DataBlade

[1] For details on the IDS 2000 and the experiences we made employing that system during the Cardio-OP project see also [9].

modules all provide support for such metadata. However, metadata applicable to several media types might be named differently, might follow different units of measurement, or might not be managed at all for certain media types. Hence, there is no homogeneous view onto media data in terms of technical metadata.

Simply gathering the different DataBlade modules in one database thus is not an adequate solution for the integrated management of multimedia data as the access to media data is by no means homogeneous. It is undesirable that the complexity of applications on top of the Media Server increases just because these applications have to cope with the different access mechanisms to media data with the heterogeneous technical metadata dependent on the particular media type.

To overcome this heterogeneity, we have designed and implemented the Media Integration DataBlade (MIB) module [10] constituting the integrated media management component of the Cardio-OP multimedia repository. It establishes a thin integration layer upon commercial, media type-specific DataBlade modules overcoming their heterogeneity by offering client applications *transparent access* to media data and a *homogeneous view on technical metadata* while still allowing access to the sophisticated media type-specific functionality of the commercial DataBlade modules, such as means for similarity search and secondary access methods. These key features of the MIB are now explained in more detail.

A.1.a *Transparent media access.* A well known solution to the problem of referencing media data held in different storage locations like file systems, web servers, binary large objects, and dedicated media servers are *locators*. A locator is a data type which abstracts from the place where media data is stored by encapsulating its storage location. A locator comes with a set of access routines. Depending on the particular storage location encapsulated in an instance of the locator, the access routines use the access methods appropriate for that storage location. Thus, it is hidden from a client application where the media data is actually stored and by which method exactly it is accessed – an application always calls the same access routines. Figure 3 illustrates the mapping of a locator-provided access routine to storage location-specific access methods with an opening request to a media object.

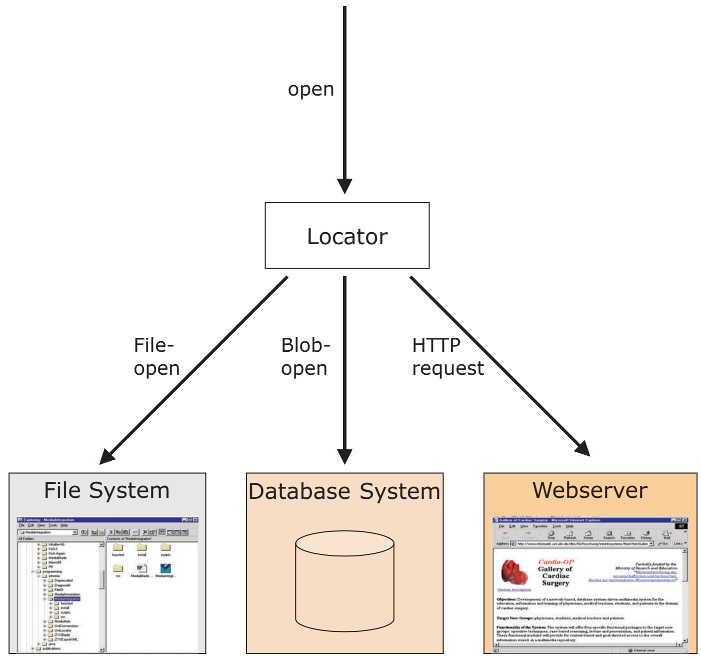

Fig. 3. The locator mechanism

Each of the commercial media type-specific DataBlade modules (namely the Informix Video Foundation DataBlade module [11], the Excalibur Image DataBlade module [12], and the Excalibur Text Search DataBlade module [13]) employed for the Cardio-OP repository provides some sort of locator data type. Unfortunately, these variants not only support different storage locations for media data but also come with different access routines which leave them incompatible with each other. Hence, none of the locator variants suits the need for transparent access to media data across the boundaries of media types.

To solve this dilemma, the MIB provides a new, *media type-independent* locator data type. It offers transparent access to an extensive set of storage locations among which are all of the storage locations supported by the commercial DataBlade modules. More precisely, these are the ORDBMS server's file system, binary large objects of the ORDBMS, web servers accessible via HTTP, and media servers supporting the Informix Virtual Storage Interface (VSI) [11], such as the Real Video Server. It can be easily extended with support for additional storage locations if desired.

To gain access to the media type-specific functionalities of the commercial DataBlade modules by the media-type independent locator, e.g., to convert an image to a different format using the Excalibur Image DataBlade, *typecasts* have been provided between the new locator type and the ones of the commercial DataBlade modules. This is necessary as the functionalities of a commercial DataBlade module of course expect media data to be referenced by the module's own locator variant.

However, not all instances of the media type-independent locator data type can be casted to all media type-specific locator variants – some of the latter might not support the particular storage location referenced. It is left to the client application to decide whether to drop support of a storage location for a certain media type in favour of multimedia functionality provided by the corresponding media type-specific commercial DataBlade module or not.

A.1.b *Homogeneous view on technical metadata.* It is desirable to manage descriptive technical metadata along with the references to media data. Descriptive technical metadata proves valuable when it comes to the retrieval of media data. For instance, a user might be interested in images of size less than 20 Kbytes and a color depth of 8 bit only while authoring a multimedia document targeted at a presentation environment characterized by limited network bandwidth and weak display capabilities. Media type-specific technical metadata come with the commercial media DataBlade modules employed as the foundation of the media management of the Cardio-OP repository. However, as outlined before, this metadata cannot be accessed homogeneously across different media types. What is needed for efficient retrieval is an integrated, homogeneous view on media types and their associated technical metadata.

The MIB creates an integration layer that provides such a homogeneous view onto media data and associated technical metadata which is illustrated in Figure 4. Resting upon the media type-specific DataBlade modules, the integration layer models each media type by a table with its columns representing technical metadata appropriate for the particular media type and referencing the media data using the MIB's locator data type. Technical metadata being shared between different media types is named and typed equally. Whenever technical metadata of a certain media type of the integration layer is also supported by the media type-specific DataBlade module employed, the table modeling the media type

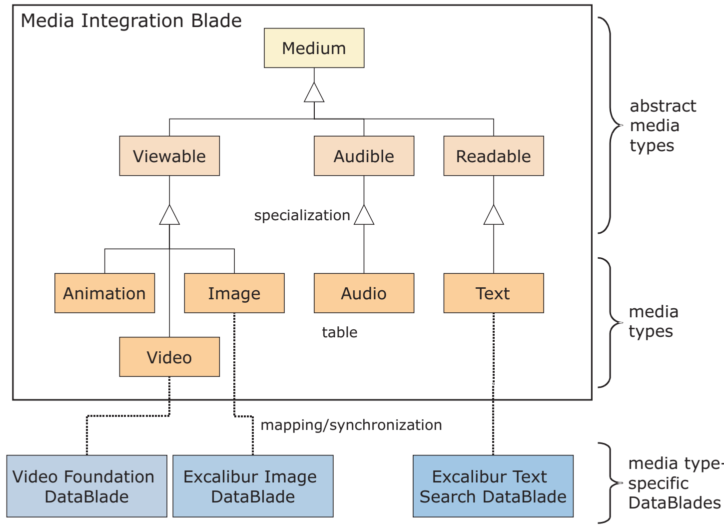

Fig. 4. Homogeneous view onto media data

is synchronized with the DataBlade module by means of triggers to assure consistency of technical metadata.

As it is reasonable to support cross-media type queries, the media types are further organized in a specialization hierarchy[2], with the root of the hierarchy modeling an abstract media type described by those technical metadata applicable to every media type and the inner nodes modeling abstract media types grouping metadata shared among closely interrelated media types. As an example of cross-media type queries, a user might be interested in viewable media data regardless whether the media data is a video, an image, or an animation.

A.2 Multimedia document management

Multimedia applications like Cardio-OP need support for the complex composition of media elements. For example, for the multimedia supported training of a bypass surgery, a video of the actual operation should be presented along with a tightly synchronized vizualization of the display of the heart-lung machine. This calls for a data model for the representation of this composition of media elements – a *multimedia document model*. A multimedia document model is employed to model the relationships between the media elements participating in a multimedia presentation. Basic requirements to multimedia

[2]The IDS 2000 supports specialization between tables.

document models are the modeling of the temporal and spatial course of a multimedia presentation and also the modeling of user interaction. However, multimedia applications have evolved: beyond the "traditional" requirements to multimedia document models, projects like Cardio-OP call for advanced support of *reuse* and *adaptation* of the multimedia content. Why do we consider these to be the new requirements to multimedia document models? As authoring of multimedia information is a very time consuming and costly task the reuse of material is definitely of high interest simply from an economical point of view. But reuse by means of "cut and paste" obviously can not be a solution, rather distinct and fine-grained reuse of multimedia content is highly demanded. Personalization and adaptation of information systems to personal needs and personal interests become more and more important (e.g., [14]). In Cardio-OP, multimedia content will be consumed by different user groups – surgeons, anesthesiologists, cardiologists, medical students, perfusionists, and patients – with different educational background and information needs.

Given these requirements, we were looking for a suitable modeling support among existing multimedia document standards. Therefore, we elaborated both the traditional and advanced requirements to multimedia document models and, endowed with this metrics, analyzed the document models HTML [15], MHEG [16], [17], HyTime [18], [19], and SMIL [20]. The detailed analysis and comparison of the models, their basic modeling concepts as well as their support for the advanced requirements shows that each of the models lacks some significant concepts and does not meet all of the requirements [21], [22], [23]. Therefore, we designed and implemented the ZYX model [24], [25] to overcome these limitations and to have a proper basis to start out from to comprehensively provide for reusability and adaptive multimedia presentations.

In the following, we, in brief, introduce the ZYX model, a formal description of which can be found in [24]. 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. The nodes of the document tree can be media elements that represent the media data but also elements that represent the temporal, spatial, and interaction relationships between their child nodes. Each presen-

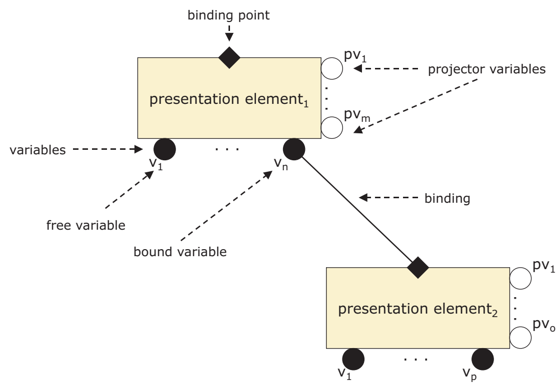

Fig. 5. Graphical representation of the basic document elements

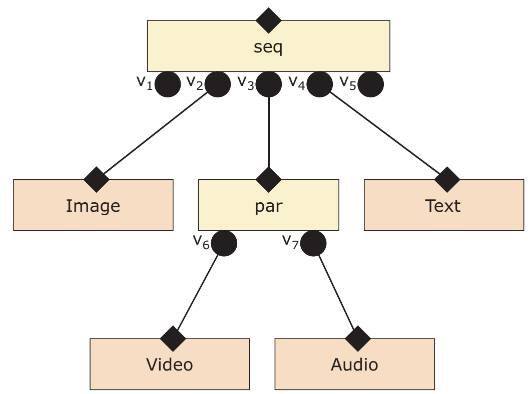

Fig. 6. Simple document tree – a ZYX fragment

tation element has one *binding point* with which it can be bound to another presentation element. It also has one or more *variables* with which it can bind other presentation elements. Additionally, each presentation element can bind *projector variables* to specify the element's layout. Figure 5 introduces the graphical representation of these basic elements of the model. The presentation elements are represented by rectangles, they form the nodes of the document tree. On top of this rectangle, a diamond represents the element's binding point. The variables are represented by the filled circles below the rectangle. The open circles on the right side of each presentation element represent the element's projector variables. The actual connections of variables and projector variables to binding points of other presentation elements are represented by edges in the graphical representation. A variable that is connected to another presentation element is called *bound* variable, those variables that are not connected are called *free* variables.

Based on the Interval Expressions [26], the model offers the primitives *seq*, *par*, *loop*, and *delay* to specify temporal interval relationships. For example, figure 6 illustrates a ZYX fragment, that temporally aligns an image, a parallel video-audio presentation and a text. With the *seq* element's binding point this fragment can be bound to another presentation element in a more complex multimedia document tree. The variables v_1 and v_5 of the fragment are still unbound. Here, a title at the beginning and a summary at the end of the sequence, can be inserted later.

Special *projector elements* provide the specification of visual and audible layout in a presentation of a multimedia document. Projector elements determine how a media element is presented. 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*.

Special focus of the development of ZYX lied on fulfilling the specific requirements of reusability and adaptation. Reuse is supported on arbitrary levels of granularity. On the level of media elements, reuse is supported by means of *selector elements*. These are presentation elements that determine which part of a media element is presented. They can be used to, e. g., select and thereby (re)use a specific part of an audio or a specific area of an image. Reuse is furthermore supported on the level of *fragments* of documents. This allows to logically partition multimedia documents and to reuse the fragments in different context. To make reuse easier and to alleviate the handling of large documents, fragments can be *encapsulated* by *complex media elements*. Then, an encapsulated fragment appears like a single presentation element in the specification tree with one binding point and possibly a set of variables. In the example in Figure 7, different presentation elements of the fragment leave variables unbound, which makes it a *template*. Nevertheless, it is possible and sensible to encapsulate this fragment by a complex media element. In that case, the free variables of the fragment are *exported* and become the variables of the complex media element which can be filled later. A complex media element constitutes a kind of black box view to a possibly complex presentation fragment. The concepts of free variables in combination with complex media elements guarantee comprehensive and workable reusability on the level of presentation fragments.

Adaptation means that the ZYX document that is delivered for presentation should best match the context of the user who requested the document. The ZYX data model provides two presentation elements for an adaptation of the document to a user profile: the *switch* element, which is similar in principle to a switch statement in a programming language, and the *query* element. While a *switch* element can be used only if all alternatives can be modeled at *authoring* time, a *query* element is evaluated at *presentation* time and dynamically replaced by the fragment best matching the metadata specified in the *query* element.

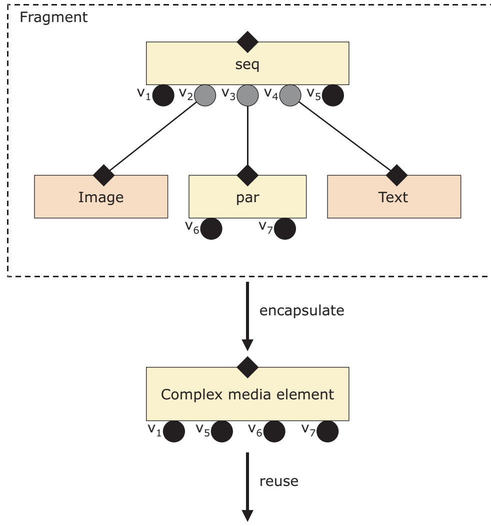

Fig. 7. Complex fragment encapsulated in a complex media element

A management component for documents following the ZYX model has been implemented as a DataBlade module for the IDS 2000 [9]. The *ZYX Blade* allows the comfortable and consistent creation and manipulation of ZYX documents and fragments inside the DBMS and provides a means to deliver a ZYX document to a client in an XML-based representation, e.g., for presentation purposes.

A.3 Metadata management

Regarding the high costs of media production, it is imperative that the content providers of the multimedia repository can thoroughly exploit their precious pieces of media data by efficient reuse in manifold contexts. The key to enable profound exploitation of the media data managed in a multimedia repository is *metadata* [27], [28], [29] describing the content as media data themselves usually encode only the presentation of content and not the information conveyed. Thus, a central building block of the Cardio-OP multimedia repository architecture is its metadata management component.

Thorough reuse of media data in manifold contexts implies efficient retrieval mechanisms. As already outlined before, retrieval may proceed along technical metadata, along metadata describing the information conveyed by media data, or along a mixture of both. Since technical metadata is managed by the media management facility, the metadata management component is focused on metadata describing the information conveyed by media data.

In order to enable efficient, content-based retrieval, several requirements to the metadata management component can be identified. First, the metadata management component must handle standardized, domain-specific vocabularies or *taxonomies* that can be employed for the description of media content. Free text descriptions of media data are obviously not a solution considering the highly varying backgrounds and contexts of the users intended to work with the Cardio-OP repository: an experienced surgeon can be expected to employ different terms for the description of media data than a student or even a patient searching for the same media data. A standardized taxonomy forces creators of media content and users looking for content to think in the same concept space. Next, it must be possible to *annotate*, i.e., associate, content with the concepts of the standardized taxonomy to describe the information conveyed. Annotation of content should take place at an adequate level of granularity: it is not very helpful to annotate a video of two hours length as a whole with a concept if that concept is relevant for a time interval of 5 seconds duration only. Rather, only the relevant interval should be annotated with the concept. Finally, *retrieval* of content according to the annotations should be enabled by the metadata management component.

We have implemented the metadata management component of the Cardio-OP multimedia repository as a DataBlade module called *OntoBlade* for the IDS 2000. In the following, we illustrate how OntoBlade addresses the requirements given above.

A.3.a *Taxonomies.* Based on the commercial COCOON DataBlade module [30], the OntoBlade provides a data model for the management of taxonomies for the standardized description of content. Figure 8 illustrates the basic constituents of the OntoBlade's data model for taxonomies with an example taken from the Cardio-OP project. The elementary entities of that data model are the *concepts* of the application domain. These and only these concepts can be used for the annotation of content. For example, in Figure 8 the concepts “chronic aortic aneurysm” and “heart disease” are part of the applications domain's taxonomy. To improve retrieval, concepts are organized in a specialization hierarchy and are further augmented with possibly ambiguous synonyms in different languages. The first reduces annotation overhead since given a query for content dealing with a certain concept, any content annotated with a direct or indirect subconcept of it also qualifies for that query. The latter is a tribute to the different origin, background, and education of the various users of the system: they are likely to use different terms in different languages[3] for the same concepts. In the given example, "chronic aortic aneurysm" is defined as a subconcept of "complication" and the German term for it is "Kreislaufstörung".

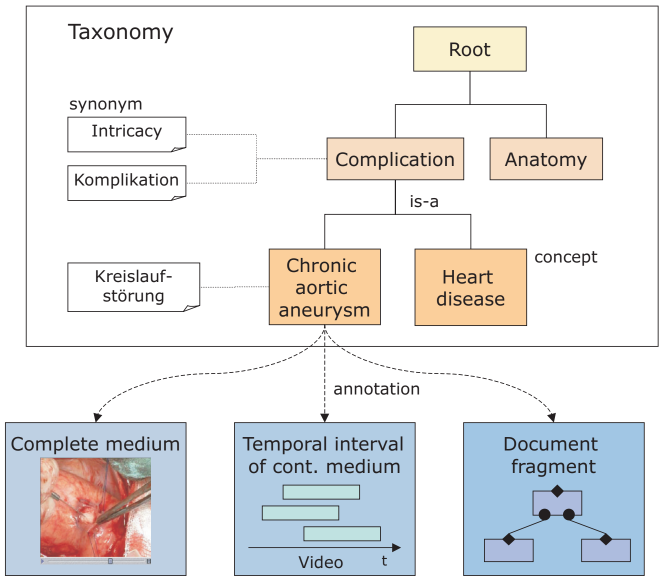

Fig. 8. Taxonomy-based annotation of multimedia content

Given that data model, the medical partners developed a taxonomy named *Cardio-OP-Data-Class* (CDC) specific to the domain of cardiac surgery [31]. Even though standardized medical taxonomies exist in large number, the domain of cardiac surgery is so specialized that they are by far too general and too less detailed to be useful for Cardio-OP. Sample tests with SNOMED, MeSH, and others revealed that only a small fraction of the concepts and terms needed to describe, e.g., a heart transplantation in full detail are provided by these vocabularies.

A.3.b *Annotation of content.* To offer adequate retrieval capabilities, the OntoBlade allows annotation of content with concepts flexibly at different levels of granularity ranging from complete pieces of content to small parts as indicated in Figure 8. Any media data

[3]The term language in this context not only subsumes natural languages like English or German but also vocabulary specific to certain communities like medical English or colloquial English.

held in the MIB, either discrete or continuous, can be associated with concepts as a whole or, in case of continuous media data, just a temporal interval of the media data. Furthermore, any fragment of a multimedia document (including complete documents) kept by the ZYX -Blade can be annotated with concepts.

A.3.c *Content-based retrieval.* For retrieval of content based on annotations with concepts, a variant of extended Boolean information retrieval [32] is provided by the OntoBlade. Boolean operators for the search for content along conjunctions and disjunctions of concepts are provided that additionally take into account specialization relationships between concepts, synonyms of concepts, and the granules of annotated content. Consider the example of Figure 9 illustrating the query for content dealing with "Intricacy" *and* "Anatomy". The video depicted is considered relevant because it is related to both concepts. There is a temporal interval directly annotated with "Anatomy" and another which is annotated with "Chronic aortic aneurysm". The latter matches the search for "Intricacy" as, recalling the taxonomy of Figure 8, "Intricacy" is a synonym for "Complication" and "Chronic aortic aneurysm" is a subconcept of "Complication". Furthermore, the retrieval takes into account that the granularity of the annotations of the example video are temporal intervals and, hence, only the intersection between the annotated intervals is returned as the retrieval result – only during that temporal interval both requested concepts are treated at the same time. For a detailed, formal definition of the content-based retrieval semantics of the OntoBlade, refer to [10].

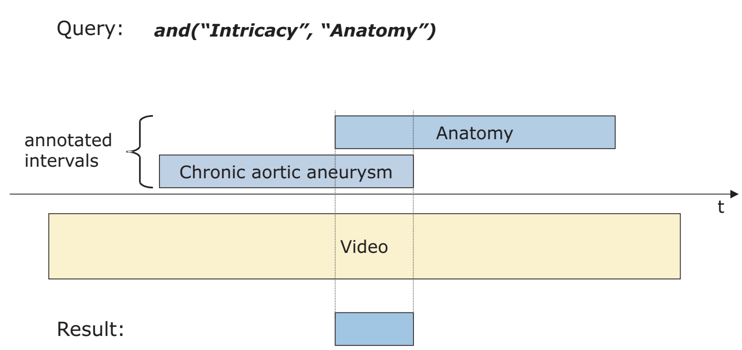

Fig. 9. Content-based retrieval of continuous media data

The OntoBlade allows to return the results of content-based retrieval in a temporary table. This enables mixed retrieval of media data according to annotations and to technical metadata by simply joining the temporary table containing the results of the search for content along annotations with the table of the MIB modeling the relevant media type with the associated technical metadata.

B. Continuous Media Streaming Facility

With the presentation of continuous media elements like video and audio when browsing the content of the repository or in the context of a multimedia presentation, a user expects a smooth and continous playback of the media element starting quickly after the request. Long startup latencies or jittering videos are undesirable. However, continuous media elements potentially have a very high data volume and the Internet protocol does not guarantee fixed bandwidths.

The solution here is *continuous streaming*. This means that after a request for a media element a stream of small units of media data is continuously sent to the client. As continuous media data is very sensitive to even small time delays during presentation, the data must be sent to the client a little ahead of presentation time, buffered locally such that the presentation client can continuously read the data from that buffer.

In the project, however, the requirements to the streaming support go beyond simply prefetching and buffering the continuous data units ahead of presentation time. As the Cardio-OP system implements an educational multimedia repository, the users are likely to heavily interact with the system. Simply buffering in a forward-oriented matter does not help if a user, e.g., jumps to another part in a video. Then all units in the buffer become invalid, the prefetching starts up again, and after some latency the presentation starts at the new presentation point. As this can be very tiring in a highly interactive usage of the system, rather quick reactions to user interactions are necessary. Additionally, the expected users of the Cardio-OP system are expected to be accessing the system from a very heterogeneous system environment. That means that the media streams are delivered over networks with different bandwidth to clients with different computing power. Therefore, it is helpful, if the stream is able to adapt to varying bandwidth in a heterogeneous system infrastructure.

Though some existing streaming technologies are able to adapt to varying bandwidths, they do not provide smooth reactions to user interactions with the continuous stream but only support forward-oriented streaming, let aside that the stream must be stored in the respective proprietary format. To meet the specific requirements described above, we developed the *MPEG-L/MRP* strategy [7], an adaptive prefetching algorithm for videos encoded in the MPEG-1 standard format to continuously deliver MPEG-1 video streams over an IP network in combination with an intelligent buffering technique that supports smooth and quick reactions to user interactions with the stream.

The Continuous Media Streaming Facility is the connection between the Media Server and the Tool Suite. It embeds the prefetching and buffering strategy into the client tool set, e.g., the multimedia presentation engine, and integrates it with the Media Server of the multimedia repository. By this the Continuous Media Streaming Facility enhances the multimedia repository by an efficient means to deliver continuous streams of interactive multimedia presentations over existing IP infrastructure trying to minimize interaction response time and optimize loading/reloading portions of a MPEG video stream.

B.1 MPEG-L/MRP

In order to support user interaction in a satisfying way, i.e., to reduce the round trip time from the client to the server after a user interaction, the MPEG-L/MRP strategy ensures that those frames that are likely to be played directly after a user interaction are available in the client's presentation buffer. Figure 10 illustrates a continuous video stream with a current presentation point p and a set bookmark b. For each of the possible interactions, play forward, fast forward, reverse, and jumping to a bookmark, the frames are shown that must be in the client's buffer to immediately continue the presentation after the interaction. Thus, these frames can be directly handed to the player to provide an extremely quick and smooth reaction to the interaction.

To achieve this, the strategy calculates *relevance values* for each MPEG-frame depending on the current presentation state. For each frame the relevance value expresses how relevant the frame is for the future interactive presentation. In this respect, the strategy is based on the L/MRP [33] strategy. The latter, however, supports only homogeneous continuous data streams like Motion-JPEG which has no practical relevance and a much lower compression rate than MPEG. Hence, we adapted the formal model in the MPEG-

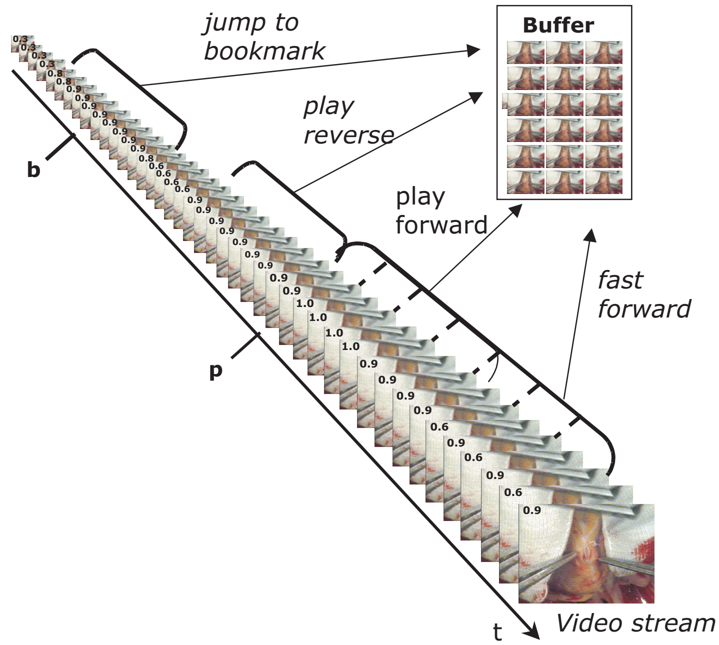

Fig. 10. Interactive continuous streaming with MPEG-L/MRP

L/MRP [7] strategy to the specific features of MPEG with its different frame types and inter-frame dependencies.

The relevance value of a frame is the result of a function of the current presentation point p and the frame's temporal distance to the presentation point p or a bookmark b. The relevance functions are monotonuously decreasing, which means that the further a frame is away from the current presentation point p and a bookmark b, the less relevant it is for the presentation.

For the calcuation of the relevance, it is important to note that with MPEG the display order in which the frames are presented is different from the bitstream order in which the frames are decoded due to inter-frame dependencies. Additionally, even though not all frames might be needed due to frame skipping, the inter-frame dependencies can cause that skipped frames are still needed for decoding needed frames. Figure 11 shows for example that a double-speed playout of an MPEG-1 video does not only need every other frame to be preloaded and buffered but also some additional frames that are needed to resolve the MPEG-specific inter-frame dependencies.

Therefore, an MPEG-1 preloading and buffering strategy should also take into account

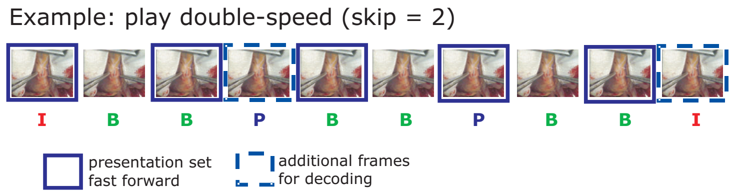

Fig. 11. MPEG-1 frame decoding dependencies for playing double-speed

the different importance of the frames for decoding and presentation. Figure 12 illustrates relevance values of the frames depending on their distance to the current presentation point and their frame type for the play forward interaction. The closer the frames are to the current presentation point the higher the relevance value is. As the different frame types are differently relevant for the stream and are differently relevant for the decoding, the most relevant I frames are preloaded before P frames and B frames. This ensures that no B frame is unnecessarily preloaded while a P frame that is needed for the B frame's decoding is missing. Also this means, that on low bandwidths the less important B and P frames are dropped first and replaced by synthetic substitute frames.

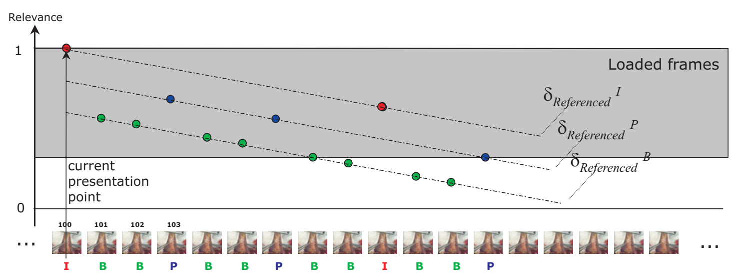

Fig. 12. The relevance function for play forward in MPEG-L/MRP

The relevance values are calculated for the different interaction types, and the formal model also specifies how these are merged into a global relevance value for each frame which then reflect both the different user interactions but also the different MPEG frame types. On the basis of these values the buffer manager prefetches the most relevant frames and purges the least relevant frames.

The MPEG-L/MRP has been implemented in Java, employing the Java Media Framework (JMF) [34]. We realized the client buffer providing an interface that meets the requirements of the JMF players such that the standard MPEG player of JMF can be connected. The client buffer prefetches the frames from a server corresponding to our strategy and delivers a correct MPEG-1 data stream to the player. In case of insufficient bandwidth, the algorithm performs an adaptation of the medium by ordering only the most relevant frames from the server. This leads to a continuous presentation at a lower frame rate if the network is overloaded. This client component is used in the entire Tool Suite for continuous MPEG delivery. On the server side, we implemented a Java program that performs the task of transferring the requested frames from the media server to the client. This MPEG-L/MRP server resides on top of the Media Integration Blade and directly reads the requested video using the MIB's media type-independent locator mechanism.

With MPEG-L/MRP embedded in the systems's architecture, a much better *Quality of Presentation* can be delivered due to smooth reactions to the expected frequent user interactions. Consequently, a much better *Quality of Information* is delivered to the user who – not to be forgotten – might have to pay for the multimedia material presented.

C. The Tool Suite

To give users a means to effectively work with the various complex data managed by the Media Server, a set of easy-to-use graphical tools covering the most common steps occurring in the multimedia production cycle is imperative. Thus, the Cardio-OP multimedia repository features a Tool Suite as an integral part. Following the architecture of Section III-C, the Tool Suite contains graphical tools supporting media annotation and retrieval (IV-C.1), authoring tasks (IV-C.2), and presentation (IV-C.3). To a large extent, these tools are designed as reusable components that applications working with the multimedia repository can exploit to quickly provide support for various basic working tasks.

C.1 Media annotation and retrieval

Fundamental and frequently occurring tasks when operating the multimedia repository are the creation and manipulation of domain-specific taxonomies, the import of raw media data to the Media Server, annotation of media data and multimedia documents with concepts of the taxonomies, and the retrieval of media data and documents according to these annotations. The Tool Suite provides several tools for the simplification of these tasks, namely the *CDC Editor*, the *Media Importer*, the *Annotation Tool*, and the *Media Browser* which we introduce in the following.

C.1.a *CDC Editor*. In the Cardio-OP project, a geographically distributed editor tool is required that allows domain experts from different hospitals to simultanously create and manipulate those subdomains of the developed medical taxonomy Cardio-OP-Data-Class (CDC) corresponding to their particular expertises. To avoid redundant or conflicting definitions, however, each author must be able to browse and search the current state of the complete taxonomy at any time. Finally, an automatically created change log is desirable to trace back the modifications performed by different authors if necessary.

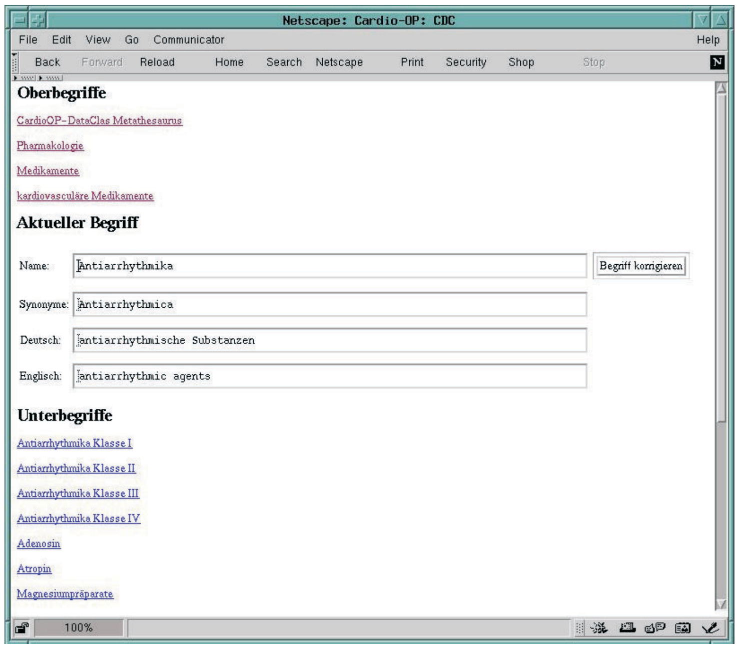

Fig. 13. CDC Edtitor

Given these requirements and in order to quickly bootstrap the taxonomy acquisition process in the beginning of the project, we developed a simple web-based solution with CGI scripts. With the CDC Editor (see Figure 13), the taxonomy maintained by the OntoBlade of the Media Server can be inspected and modified using a standard Web browser. In future, we intend to replace this ad-hoc solution by a more sophisticated taxonomy editor tool, such as OntoEdit [35] or Protégé-2000 [36].

C.1.b *Media Importer.* One of the most important tasks during the multimedia production process is the import of newly acquired media data into the multimedia repository. For that purpose, the repository offers the *Media Importer* (see Figure 14), a tool which introduces new media data to the Media Server: either by uploading the media data from the file system of the client running the tool to the Media Server, or by addressing the media data using the locator mechanism of the Media Integration Blade, in case that the storage location of the media data can be reached by a locator.

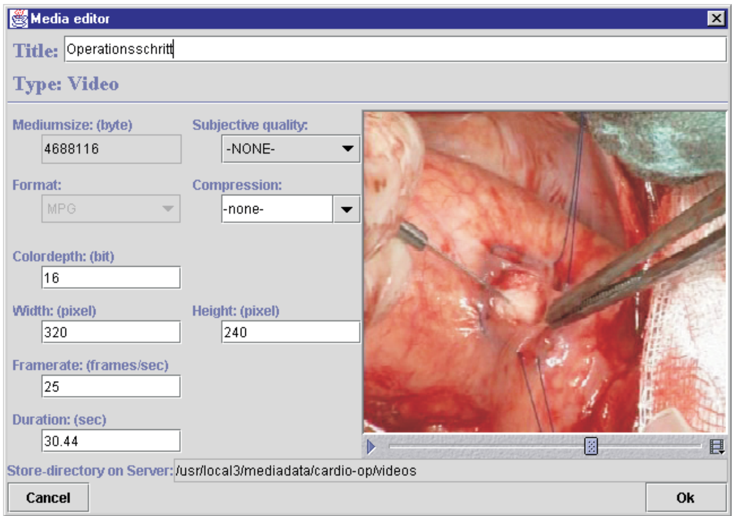

Fig. 14. Media Importer

During import, technical metadata applicable to the media type is acquired and associated with the imported media data in the Media Integration Blade. As far as possible, the import tool extracts the technical metadata automatically. However, the user is free to manually change and add technical metadata at any time. The Media Importer has been implemented in Java using the Java Media Framework and Swing. To reuse this component in other Java applications that need to import media data to the Media Server, it has been developed as an easy-to-use panel component of the Swing GUI framework.

C.1.c *Annotation tool.* Given the ability to populate the Media Server with media data and to provide a taxonomy for the description of content, a means must be provided for the content providers to enrich their precious pieces of media data with descriptional annotations using concepts of the taxonomy for further exploitation. This is the purpose of the *Annotation Tool*. The Java-based Annotation Tool constitutes a graphical frontend enabling the annotation of discrete and continuous media data as well as the annotation of multimedia documents and fragments of multimedia documents with concepts from the taxonomy.

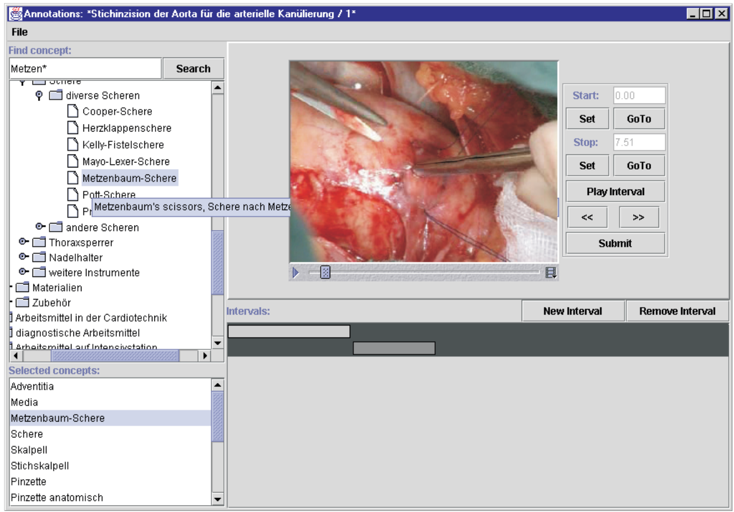

Fig. 15. Annotation Tool

Figure 15 gives an impression of the Annotation Tool. It depicts a screenshot of the annotation of a video. The video to be annotated is presented in a media player in the center of the annotation dialogue. On the left hand side, the taxonomy as managed by the OntoBlade is displayed in a tree view for exploration. The identifiers of the concepts form the nodes of the tree view while the synonyms applicable to a certain concept are shown in a tool tip appearing when pointing to that concept with the mouse. Concepts may be selected (bottom left) directly from the tree view. Moreover the user may search for concepts by entering the identifier or a synonym of the concept desired (top left). When annotating discrete media data and multimedia documents, the selected concepts refer to the medium as a whole. When annotating continuous media data (as it is the case in Figure 15), the concepts rather refer to a time interval of the continuous media data. These annotation intervals are gathered in a timeline view in the lower half of the annotation dialog. This view allows to select the current annotation interval (the one which is highlighted), to define new annotation intervals, and to remove already existing ones. The starting point and the end point of a time interval may also be changed by simply dragging the boundaries of the interval along the timeline. In addition, the currently selected annotation interval can be presented in the media player and its starting and end points can be alternatively adjusted using the VCR-like controls to the top right.

C.1.d *Media Browser*. It is a goal of the Cardio-OP project that the users of the multimedia repository can thoroughly exploit the valuable content managed by the Media Server by reusing them in as many contexts as possible. Thus, a frontend is necessary enabling users to effectively keep track of and to retrieve the various media data and multimedia documents managed by the Media Server. The *Media Browser* constitutes an intuitive frontend for sophisticated retrieval.

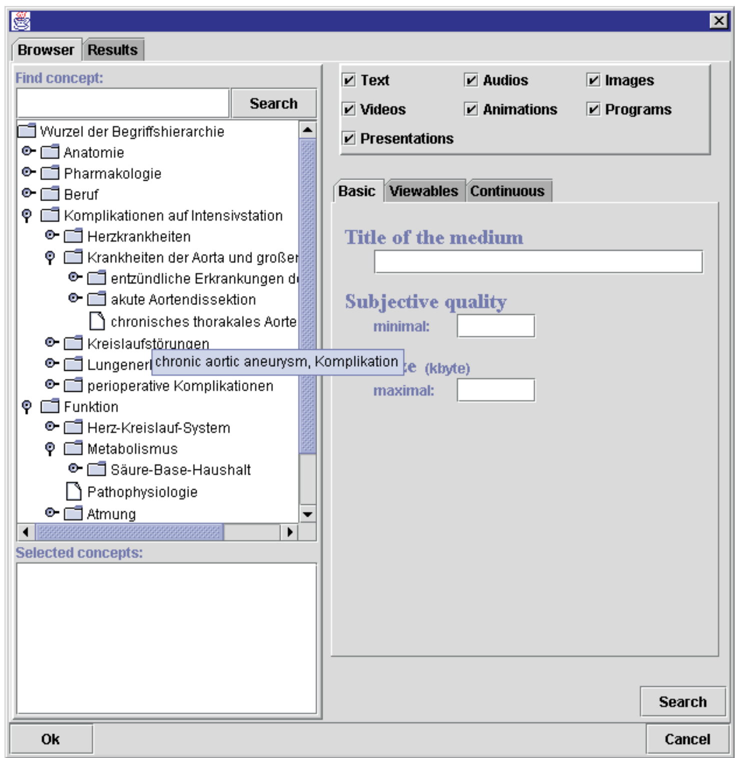

Fig. 16. Media Browser

With the help of the Media Browser, content can be retrieved according to type, technical metadata, and annotations. At the top right of the browsing dialogue (see Figure 16 for a screenshot), the media types the user is interested in may be specified. Depending on the media types selected, the Media Browser allows to further narrow down the search by specifying constraints on technical metadata that media data has to fulfil in order to qualify for the retrieval result. This is done using the input form to the right. Finally, the retrieval result can be constrained to content annotated with those concepts of the taxonomy that have been selected using the tree view to the left. The tree view is similar to the one encountered with the Annotation Tool. Selection of more than one concept is interpreted as the conjunction of the selected concepts.

After initiating the retrieval process using the search button to the lower right, the Media Browser transforms the specifications made by the user to a query that is executed at the Media Server. The query exploits the functionality provided by the MIB and the OntoBlade for content-based retrieval as we have outlined in Section IV-A. As its result, the query returns references to those discrete media data, time intervals of continuous media data, and multimedia documents that fulfil the retrieval constraints specified by the user.

Similar to the Media Importer, the Media Browser has been implemented as a reusable panel component of the Swing GUI framework. Any Java application that needs to provide access to content managed by the Media Server can thus employ the Media Browser for retrieval tasks with only little efforts.

C.2 Authoring

Having the tools of the Cardio-OP multimedia repository for the management of media data and associated metadata at hand, experts of the domain of cardiac surgery must be given the ability to compose complex multimedia content from the material available.

In this respect, it must be considered that the creators of content are *domain* experts and not *IT* experts. Thus, it is not possible to simply employ generic, off-the-shelf multimedia authoring solutions, e.g., Macromedia Director, as these solutions make great demands on the technical and programming knowledge of an author – surgeons neither have the time nor the will to become acquainted with multimedia authoring at the level of "multimedia programming." Instead, higher level, *domain-oriented* authoring facilities should be offered to domain experts, allowing them to compose appealing multimedia presentations for their particular field of expertise thinking in concepts they are accustomed to without bothering too much about technical details.

Additionally, it must be considered that commercial authoring systems often neglect or even ignore the need for reuse of media data in different contexts and settings. This makes thorough exploitation of expensively produced content in different contexts difficult to achieve with these tools. However, this is a central requirement of the Cardio-OP project.

Consequently, the Tool Suite of the Cardio-OP multimedia repository provides specialized tools for the authoring of multimedia content in the domain of cardiac surgery on a high level of abstraction promoting extensive reuse of the media data managed by the Media Server. Our experiences made during the Cardio-OP project have shown that cardiac surgeons have been able to effectively produce large amounts of multimedia content using these tools after only a small period of familiarization. The tools, namely the *Cardio-OP Multimedia Book Editor (COMEd)* and the *Authoring Wizard*, are described in in more detail in the following.

C.2.a *COMEd — Cardio-OP Multimedia Book Editor.* In the Cardio-OP project, an essential part of the multimedia content the multimedia repository has to manage is the multimedia book on operative techniques. Though it is considered a *multi*-media book, its primary and leading medium is – similar to traditional textbooks – simple text. It is hierarchically composed of chapters, sections, subsections, etc. Furthermore, well-known cross references from one division of the book to another – which might also be called hyperlinks in the context of an electronic book – are frequently occurring. To fully exploit the advanced possibilities of an electronic book, media objects like video clips, photographs, 3D animations, etc., which illustrate the textual explanations, as well as additional graphical objects like arrows, shapes, and so on, which highlight particular details on demand, might be associated with text passages.

Somewhat beyond the scope of a conventional textbook are *tip* and *cave*[4] paragraphs, as they are called, which are not part of the main text flow. Rather, they are intended to be read "asynchronously" – similar to footnotes or additional remarks – and on demand only.

Completely outside the realm of traditional printed books is the fact that the Cardio-OP multimedia book is actually a union of three different books on cardiac surgery for three different audiences: students (level 1), interns (level 2), and surgeons (level 3). Typically,

[4]from latin *cavere*, take care, be careful

but not necessarily, the content of one level is completely contained in the higher levels, i. e., the students book is approximately a subset of the interns book which is in turn subsumed in essence by the surgeons book.

Given these characteristics of the multimedia book on operative techniques, an intuitive tool for writing a kind of "storyboard" for its contents is needed containing a verbal description of the "drama" (the textual part of the multimedia book) as well as "stage directions" specifying when particular "actors" (media objects) should appear on the scene. An ideal tool for writing the storyboard for the multimedia book should on the one hand offer a user interface similar to the one of a WYSIWYG word processor (like, e. g., Microsoft Word or Adobe FrameMaker) since the domain experts (surgeons, in the case of Cardio-OP) are familiar with such tools. On the other hand, such a tool must provide adequate support for the advanced, multimedia features mentioned before.

To fulfill these requirements without needing to implement a specialized tool early in the project, we have decided to employ an "upper class" SGML editor, in particular FrameMaker+SGML [37].

The editor allows to associate an SGML *document type definition* (DTD) with *formatting instructions* conveying to the author the impression of editing a conventional text document (cf. Figure 17). We have created a DTD for the storyboard of the multimedia book on operative techniques and appropriate formatting instructions. By maintaining a context sensitive catalogue of the elements of the DTD, FrameMaker+SGML guides the author during the editing process of the storyboard (cf. Figure 17, small window on the right).

The result of the storyboard-authoring are SGML files, which are constrained to be also *XML*-compatible. Using a publically available XML parser, the hierarchical document structure is extracted, converted to ZYX fragments, and finally imported into the Cardio-OP multimedia repository. These fragments form the basis for the domain-based multimedia authoring in subsequent steps.

C.2.b *Authoring Wizard.* Given the need for a high-level, domain-oriented authoring environment enabling the project's medical experts to compose multimedia presentations, our partners at FAW have developed the Authoring Wizard [38]. This Java-based author-

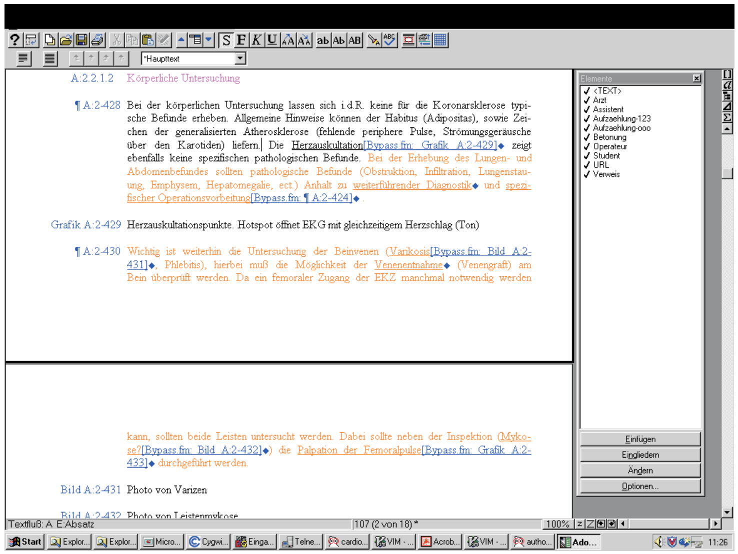

Fig. 17. COMEd

ing environment is specifically intended for the composition of multimedia presentations for the instructional applications: the multimedia book on operative techniques, the module for the training of decision making during operations, the module for patient information, and the module for multimedia lectures. Coupled tightly with the 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 thereby promoting reuse of expensively produced content in a variety of different contexts.

The basic paradigm followed by the Authoring Wizard for the creation of multimedia presentations at a high level of abstraction is inspired by wizard technology frequently occurring in office suites and programming environments: depending on the type of multimedia presentation to create, the Authoring Wizard guides the author through the particular authoring steps and offers dialogues specifically tailored to the needs of each step.

Exemplarily, Figure 18 gives an impression of the Authoring Wizard when used to author the multimedia book on operative techniques. Using the tree view on the left, an author can browse through the hierarchical structure of the book as it is defined by the storyboard created with COMEd. When a section of the book is selected for authoring, a dialogue appears (on the right) giving a schematic impression of how the section will look like when presented. For a given educational level, i.e., students, interns, or surgeons, the text of the

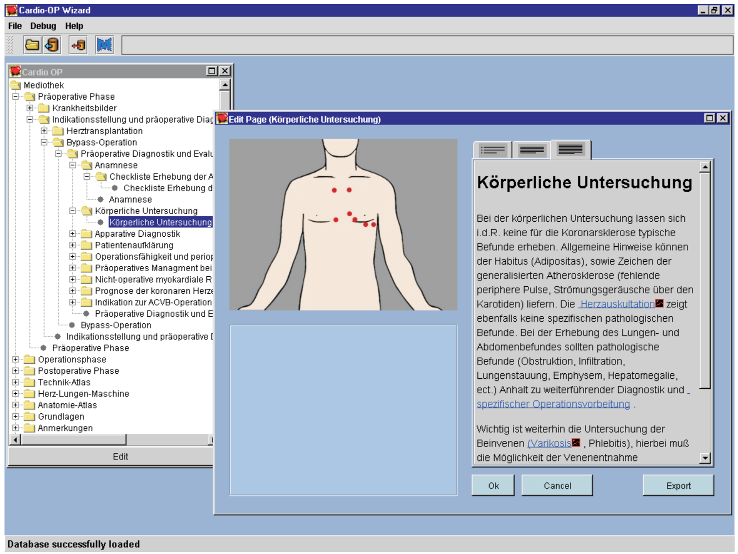

Fig. 18. Authoring Wizard

section being authored is displayed in the right half of that schematic view as defined with COMEd[5]. The educational level displayed can be altered using the three tabs above the text. The left half of the schematic view contains two rectangular areas as placeholders for media data: the upper left is used to represent media data associated with a selected text passage (showing a schematic image of the thorac in the case depicted) while the lower left is used to represent additional media data for further clarification. The media data represented by the placeholders is selected from the media available at the Media Server using the Media Browser.

An author has plenty of further editing options: for each educational level, he or she can define hotspots relating areas of interest of visual media with additional information, define various types of arrows pointing to areas of interest, define applicable background sounds such as background speakers, etc.

Once the author finishes editing the presentation type-specific dialogues, the editing results are translated to an equivalent ZYX representation and stored at the ZYX Blade of the Media Server. Then the presentation is available for rendering using the Presentation Engine (see Section IV-C.3) or for further editing with the Authoring Wizard in additional authoring steps.

[5]Note that the text is identical to the one edited with COMEd as depicted on Figure 17.

C.3 Presentation

COMEd and the Authoring Wizard provide the domain experts of the Cardio-OP project with an infrastructure for the composition of sophisticated multimedia content for different instructional applications. Additionally, an infrastructure must be provided conducting the presentation of these compositions. As already mentioned above, the compositions are ZYX multimedia documents and stored at the Media Server employing the ZYX Blade. Hence, the provision of a presentation infrastructure for these compositions effectively means the provision of a ZYX presentation environment.

Basically, two approaches for the realization of a presentation environment for ZYX multimedia documents can be identified. On the one hand, *existing presentation environments*, e.g., HTML browsers, could be reused. This approach encompasses the transformation of ZYX documents to the document model on which the presentation environment used is based. On the other hand, one could implement a *dedicated ZYX presentation software* that is able to conduct the multimedia presentations directly on the basis of the ZYX model. Since ZYX is very expressive and supports a rich multimedia functionality, it has proven difficult to generically transform ZYX documents to documents of more widespread multimedia document models such as HTML, SMIL, and MHEG-5 without losses [23], [21]. Thus, we have opted for the second approach and have implemented a dedicated ZYX presentation software, the *Presentation Engine*.

The Presentation Engine, based on Java and the Java Media Framework, conducts the presentation of an arbitrary ZYX document. Directly interacting with the Media Server, the Presentation Engine receives the ZYX document to present in an XML representation from the ZYX Blade. The Presentation Engine parses this XML representation of the ZYX document, calculates a prefetch schedule for the media data referenced in the document, and begins to prefetch the media data from the Media Integration Blade (MIB) through the media type-independent locators according to this schedule. The discrete media data is delivered directly to the Presentation engine, the delivery of referenced MPEG-1 videos is realized via the Continuous Streaming Facility, i.e., the MPEG-L/MRP client buffer strategy. Once the prefetching of the media data has progressed such that a smooth presentation of the ZYX document is feasible, the Presentation Engine commences its presentation. While presenting, the Presentation Engine enforces the temporal and spatial constraints between the media data as specified in the ZYX document, constantly prefetches further media data as they become relevant for presentation, reacts to user interactions, and, depending on the user context, chooses between presentation variants that might have been specified using the advanced adaptation primitives of ZYX.

Figure 19 gives a screen shot of the Presentation Engine while presenting a section of the multimedia book on operative techniques to a consumer[6].

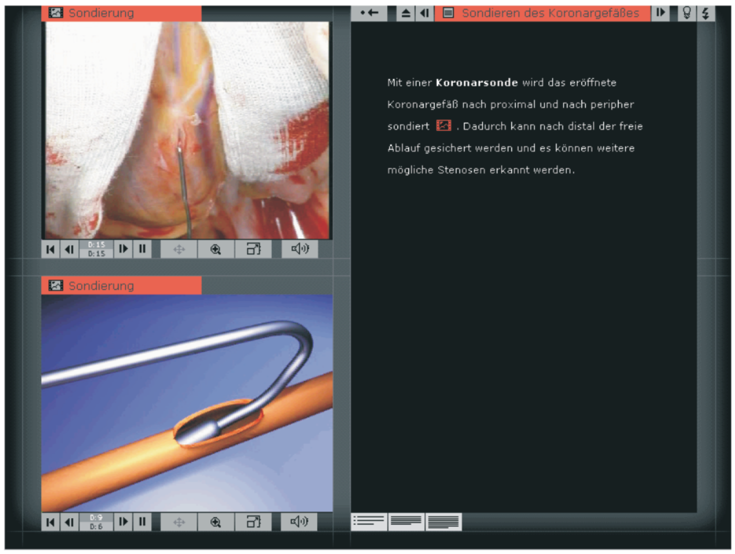

Fig. 19. Presentation Engine

The Presentation Engine has been implemented as a reusable panel component of the Swing GUI framework. Any Java application can thus integrate the presentation of ZYX multimedia documents managed at the Media Server into its GUI with only little efforts.

D. *Support of the multimedia producer-consumer cycle by the Cardio-OP multimedia repository*

Having introduced the architecture of the Cardio-OP multimedia repository and its variety of system components, we reflect on how these components relate to the multimedia

[6]The reader might recognize several basic constituents of a section of the multimedia book on operative techniques from the schematic view of the Authoring Wizard depicted in Figure 18: the main text to the right, a video related to the main text to the top left, and an animation that further illustrates the operative technique currently shown by the video to the bottom left.

producer-consumer cycle introduced in Figure 1 on page 106 and illustrate this in Figure 20. Thereby, we discuss the genericity of our solutions.

The heart of the multimedia producer-consumer cycle, the central Multimedia Storage and Management, is realized by the Media Server. The integrative management of media data, meta data, and multimedia document is provided by the particular DataBlade modules of the Media Server: the Media Integration Blade, the OntoBlade, and the ZYX Blade, respectively. These components are independent of the application domain: the MIB is generic as long as the technical metadata provided by the homogeneous view on the different media types suffices the respective application domain. The ZYX Blade is universal by construction as it implements the application-independent ZYX multimedia document model. Finally, the OntoBlade is domain-independent, as the taxonomies it hosts are not hardwired – specific taxonomies for arbitrary application domains can be managed.

The first step of the producer-consumer cycle is the media acquisition. This is where the tool support of the multimedia repository begins. Though the actual acquisition of the media data is beyond the scope of the tool suite and relies on commercial media acquisition and editing tools, the Media Importer allows to insert the suchlike acquired media and their technical technical metadata into the system.

For the purpose of Media Value Adding, the next step in the producer-consumer cycle, we support the annotation of content with concepts taken from a taxonomy. Therefore, we provide the CDC Editor to create and maintain taxonomies. With the Annotation Tool, the experts in the field find a tool to easily enrich the (continuous) media data by associating concepts from the taxonomy with them. Both tools operate on the Media Integration Blade and the OntoBlade at the Media Server.

Multimedia Browsing and Retrieval is realized by the *Media Browser* which allows to search the repository for the media along technical metadata and annotations.

The tools mentioned so far constitute graphical front-ends to the Media Integration Blade and OntoBlade and are thus similar in terms of genericity.

For the task of Multimedia Authoring, we provide only tools that support the authoring of the instructional applications defined by the Cardio-OP project. In particular, these are the COMEd and the Authoring Wizard. This solution is, intentionally, not generic. It intends to provide support for high-level authoring in the domain of cardiac surgery enabling *domain* experts to compose sophisticated multimedia presentations without being bothered with generic though tedious "multimedia programming." As COMEd is based on a generic SGML editor, however, the effort to adapt it to a similar application domain (e. g., writing a multimedia book on aeroplane maintenance) is expected to be moderate.

Multimedia Delivery comprises the delivery of the multimedia assets from the Media Server to the client tools. For multimedia documents the ZYX Blade provides functionality for the delivery of the multimedia documents in an XML-based format. This transport format only depends on the ZYX model, which is domain-independent, and is thus domain-independent as well. Whenever a client tool needs the delivery of a video the MPEG-L/MRP buffer strategy is employed for continuous streaming. As MPEG-L/MRP is capable of streaming any MPEG-1 video it constitutes a generic solution for multimedia delivery.

The multimedia production cycle is completed with the Multimedia Presentation Engine that presents arbitrary ZYX documents. Again, as ZYX is application-independent, the presentation engine is generic.

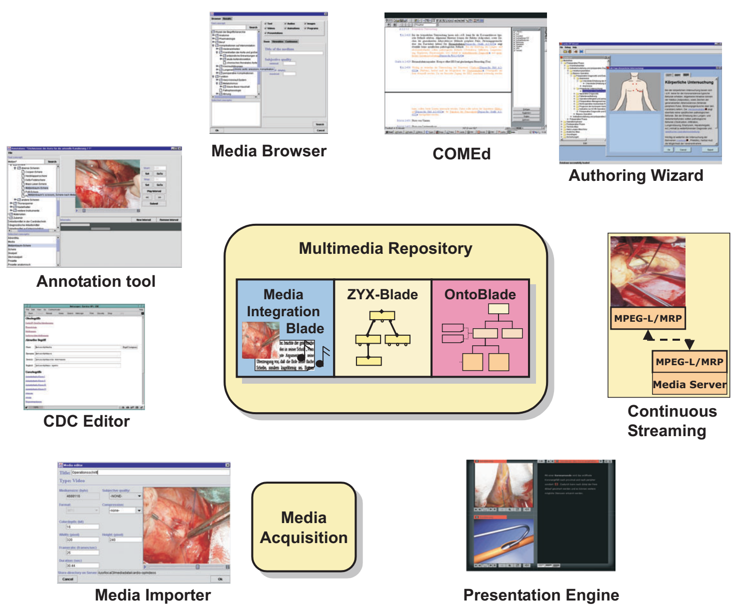

Fig. 20. Support of the Multimedia Production Cycle by the Cardio-OP multimedia repository

Summarizing we can say, that the goal to develop a multimedia repository that can be applied to different application domains is, apart from the domain-based authoring which is domain-dependent by nature, is attained by our solution.

V. Conclusion

In this paper, we presented our approach to a database-driven, network-based multimedia information system. We presented the enabling technologies and tools that provide an integrated and coherent support for the multimedia production cycle.

The approach is not limited to the medical application domain, but rather constitutes a generic approach for generating, delivering, and consuming multimedia content.

We have further intensified our research in the area of metadata management. Recently, several promising standardization efforts have reached a mature state that facilitate the extensive standardized exchange of rich metadata for multimedia content: MPEG-7 [39], RDF [40], and Topic Maps [41]. These standards will form the infrastructure of the so-called Semantic Web that aims to enable meaningful access to the vast amounts of contents lying idle in the World Wide Web today. We are currently investigating on a standard-conformant metadata management component for a multimedia repository. Standard conformance will allow to integrate the contents managed in a multimedia repository with the Semantic Web. Thereby, multimedia repositories could become a major technological foundation of the Semantic Web.

The wizard-supported authoring is just the beginning of an ongoing renewal of the authoring process itself. Manual authoring with the existing tools still needs too much expertise in using the tools and bothers the experts in the field too much such that they still refuse the effort to produce multimedia content. We are working on support of template-based authoring and (semi-)automatic generation of multimedia presentations to let the users concentrate on their expertise, their knowledge in the field.

References

[1] W. Klas, C. Greiner, and R. Friedl, "Cardio-OP — Gallery of Cardiac Surgery," in *Proc. of IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, Florence, Italy, June 1999, IEEE Computer Society.

[2] T. C. Rakow, E. J. Neuhold, and M. Löhr, "Multimedia Database Systems – The Notions and the Issues," in *Proc. of Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'95)*, Dresden, Germany, March 1995.

[3] W. Klas and K. Aberer, "Multimedia and its Impact on Database System Architectures," in *Multimedia Databases in Perspective*, P. M. G. Apers, H. M. Blanken, and M. A. W. Houtsma, Eds. Springer, London, 1997.

[4] S. Boll, W. Klas, and M. Löhr, "Integrated Database Services for Multimedia Presentations," in *Multimedia Information Storage and Management*, S. M. Chung, Ed. Kluwer Academic Publishers, Dordrecht, 1996.

[5] Kingsley C. Nwozu, P. Bruce Berra, and B. Thuraisingham, Eds., *Design and Implementation of Multimedia Database Management Systems*, Kluwer Academic Publishers, 1996.

[6] D.A. Adjeroh and K.C. Nwosu, "Multimedia Database Management – Requirements and Issues," *IEEE MultiMedia*, vol. 4, no. 3, 1997.

[7] S. Boll, C. Heinlein, W. Klas, and J. Wandel, "MPEG-L/MRP: Adaptive Streaming of MPEG Videos for Interactive Internet Aplications," in *Proc. of the 6th International Workshop on Multimedia Information Systems 2000 (MIS'00)*, Chicago, Illinois, October 2000.

[8] Informix Corp., "Informix Dynamic Server 2000," Website, URL: http://www.informix.com/ids2000/, 2000.

[9] S. Boll, W. Klas, and U. Westermann, "Exploiting OR-DBMS Technology to Implement the ZYX Data Model for Multimedia Documents and Presentations," in *Proc. of Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'99)*, Freiburg, Germany, March 1999.

[10] U. Westermann and W. Klas, "Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets," in *Proceedings of the First Intl. Workshop on Multimedia Intelligent Storage and Retrieval Management (MISRM'99)*, Orlando, Florida, October 1999.

[11] Informix Corp., *Informix Video Foundation DataBlade Module User's Guide, Version 2.0*, Informix Press, Menlo Park, California, 1999.

[12] Informix Corp., *Excalibur Image DataBlade Module User's Guide, Version 1.2*, Informix Press, Menlo Park, California, 1999.

[13] Informix Corp., *Excalibur Text Search DataBlade Module User's Guide, Version 1.2*, Informix Press, Menlo Park, California, 1999.

[14] D. C. A. Bulterman, "User-centered Abstractions for Adaptive Hypermedia Presentations," in *Proc. ACM Multimedia'98*, Bristol, UK, September 1998.

[15] D. Raggett and A. Le Hors and I. Jacobs, "HTML 4.0 Specification," W3C Recommendation, World Wide Web Consortium (W3C), URL: http://www.w3.org/TR/1998/REC-html40-19980424, April 1998.

[16] ISO/IEC JTC 1/SC 29/WG 12, "ISO/IEC 13522-5:1997 Information technology – Coding of multimedia and hypermedia information – Part 5: Support for base-level interactive applications," International Standard, International Organization for Standardization (ISO), April 1997.

[17] ISO/IEC JTC 1/SC 29/WG 12, "ISO/IEC 13522-6:1998 Information technology – Coding of multimedia and hypermedia information – Part 6: Support for enhanced interactive applications," International Standard, International Organization for Standardization (ISO), October 1998.

[18] ISO/IEC JTC 1/SC 34/WG 3, "ISO/IEC 10744:1997 Information technology – Hypermedia/Time-based Structuring Language (HyTime)," International Standard Edition: 2, International Organization for Standardization (ISO), September 1997.

[19] S. R. Newcomb, N. A. Kipp, and V. T. Newcomb, ""HyTime" – The Hypermedia/Time-Based Document Structuring Language," *Communications of the ACM*, vol. 34, no. 11, November 1991.

[20] P. Hoschka and S. Bugaj and D. Bulterman and others, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification," W3C Recommendation, World Wide Web Consortium (W3C), URL: http://www.w3.org/TR/1998/REC-smil-19980615, June 1998.

[21] S. Boll, W. Klas, and U. Westermann, "Multimedia Document Formats — Sealed Fate or Setting Out for New Shores?," in *Proc. of IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, Florence, Italy, June 1999.

[22] S. Boll, W. Klas, and U. Westermann, "A Comparison of Multimedia Document Models Concerning Advanced Requirements," Technical Report - Ulmer Informatik-Berichte Nr. 99-01, University of Ulm, Germany, URL: http://www.ifs.univie.ac.at/westermann/publications/TR99-01.pdf, February 1999.

[23] S. Boll, W. Klas, and U. Westermann, "Multimedia Document Formats — Sealed Fate or Setting Out for New Shores?," *Multimedia - Tools and Applications*, vol. 11, no. 3, 2000.

[24] S. Boll and W. Klas, "ZYX — A Multimedia Document Model for Reuse and Adaptation," *IEEE Transactions on Knowledge and Data Engineering*, 2001.

[25] S. Boll and W. Klas, "ZYX — A Semantic Model for Multimedia Documents and Presentations," in *Proc. of the 8th IFIP Conference on Data Semantics (DS-8): "Semantic Issues in Multimedia Systems"*, Rotorua, New Zealand, January 1999.

[26] A. Duda and C. Keramane, "Structured Temporal Composition of Multimedia Data," in *Proc. IEEE International Workshop on Multimedia- Database-Management Systems*, Blue Mountain Lake, August 1995.

[27] V. Kashyap, K. Shah, and A. Sheth, "Metadata for Building the Multimedia Patch Quilt," in *Multimedia Database Systems: Issues and Research Directions*, S. Jajodia and V.S. Subrahmanian, Eds. Springer Verlag, London, 1995.

[28] K. Curtis, P.W. Foster, and F. Stentiford, "Metadata - The Key to Content Management Services," in *Proceedings of the 3rd IEEE Meta-Data Conference*, Bethesda, Maryland, Apr. 1999.

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

[30] dimedis, *The COCOON DataBlade Module Users Guide*, dimedis GmbH, Cologne, Germany, 1998.

[31] R. Friedl, M. Preisack, M. Schefer, W. Klas, et al., "Cardio-OP: an Integrated Approach to Teleteaching in Cardiac Surgery," in *Proc. of Medicine Meets Virtual Reality 2000 (MMVR)*, Amsterdam, Netherlands, August 2000.

[32] G. Salton, E. A. Fox, and H. Wu, "Extended Boolean Information Retrieval," *Communications of the ACM*, vol. 26, no. 11, 1983.

[33] F. Moser, A. Kraiß, and W. Klas, "L/MRP: A Buffer Management Strategy for Interactive Continuous Data Flows in a Multimedia DBMS," in *Proc. of 21th International Conference on Very Large Data Bases (VLDB'95)*, Zürich, Switzerland, September 1995.

[34] Sun Microsystems Inc., "Java Media Framework 2.0," Website, URL: http://java.sun.com/products/java-media/jmf/2.0, 1999.

[35] A. Maedche, H.P. Schnurr, S. Staab, and R. Studer, "Representation Language-Neutral Modeling of Ontologies," in *Proc. of the German Workshop "Modellierung-2000"*, Koblenz, Germany, April 2000.

[36] N.F. Noy, M. Sintek, S. Decker, et al., "Creating Semantic Web Contents With Protégé-2000," *IEEE Intelligent Systems*, vol. 16, no. 2, 2001.

[37] Adobe Systems Inc., "FrameMaker+SGML," Website, URL: http://www.adobe.com/products/framemaker/prodinfosgml.html, 1999.

[38] C. Greiner and T. Rose, "A Web Based Training System for Cardiac Surgery: The Role of Knowledge Management for Interlinking Information Items," in *Proceedings of The World Congress on the Internet in Medicine – MEDNET 98*, London, November 1998.

[39] ISO/IEC JTC 1/SC 29/WG 11, "Information Technology – Multimedia Content Description Interface – Part 1: Systems," ISO/IEC Final Draft International Standard ISO/IEC FDIS 15938-1, International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), Nov. 2001.

[40] O. Lassila and R.R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification," W3C Recommendation, World Wide Web Consortium (W3C), http://www.w3.org/TR/1999/REC-rdf-syntax-19990222, Feb. 1999.

[41] ISO/IEC JTC 1/SC34/WG3, "Information technology – SGML Applications – Topic Maps," ISO/IEC International Standard ISO/IEC 13250, International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), Oct. 2000.