



# **An Environment for Processing Compound Media Streams**

## **- Work in progress paper V 2.0 -**

B. Feustel, A. Kárpáti, T. Rack, and T.C. Schmidt

Computer Centre, Fachhochschule für Technik und Wirtschaft Berlin

### **Preface**

Today's standards of internet-connected computers provide easy, intuitive access to multimedia information documents within classrooms as well as students homes and thereby confront students as well as teachers with a new paradigm of knowledge transfer: Not only the offer of an unfiltered totality of the present (rapidly changing) knowledge requests for continuous (network) access, but also a formerly unknown multitude of presentation methods to the lecture hall or - in the framework of teleteaching - to students homes has come around. Nothing since the invention of blackboard and chalk, we are tempted to claim, has revolutionized teaching in a more fundamental way than networked multimedia.

Consequently web-based teleteaching and distant learning offers are by now seriously considered parts of the educational system and gain increasing importance. But by preparing educational applications the insufficiency of approaches purely grounded on html-style technologies becomes more and more evident: Design and maintenance of a website approximately reflecting the complexity of an interactive online course is on the one hand an experience of little practicability. On the other hand information streams formed of time-based media or continuously online processed data hardly incorporate into a stateless presentation layer. Consequently a growing awareness for the demand of better information models can be presently observed within the community of educational computing [1],[3].

Learning modules request for a coherent design of interrelated portions of information being at the same time subject to structural subdivisions regarding thematic aspects such as topic, subtopic, related field etc., didactic classifications concerning complexity, order, relevance to the objective, etc., presentational attributes like positions in space and time, display contexts, ... and finally meta information regarding format, author, access rights etc. The meaningful shaping of such structural overlays belongs to the author. Therefore a desirable information system not only should exhibit capabilities of embedding its contents into flexible structuring but also needs to strongly support the process of authoring in accordance to its abilities, unique points of source editing being the most prominent feature under request.

Of equal relevance appears the support of multimedia data. Different types of media such as text, images, animations interplaying with time-based material i.e. audio, video or online data processing request for an individual specialized treatment which for the non technical oriented author is hard to fulfil. In recent times it has been widely understood that the preparation of qualitative advanced multimedia material ranges far beyond the scope



of individual lecturers. Facing the demand for good multimedia supplements in teaching on the one hand and recognising the difficulties in the production of such material a 'marketplace'-type idea of exchange and reuse appears quite natural in multimedia supported teaching.

Teaching has to account for perception being a time-dependent process. The important notion of time in teaching is one major reason for drawing a lot of attention in recent research works to World Wide Web techniques which distribute multimedia documents with temporal and spatial relations. In addition the growing demand for synchronized handling of time-based media such as video and audio serves as a general motive for introducing temporal aspects to the Web. Finally, streaming data sources invent a new level of scalability by accounting for transport timing and therefore rapidly gain quantitative importance throughout the internet.

In this paper we present ongoing work on an environment grounded on an object oriented multimedia information model. Residing in a database management system media objects can combine to form complex documents by means of an active document structuring, allowing for temporal and event-type inter-object relations. The environmental basis is employed for the Media Objects in Time time-based runtime environment as well as in further teaching applications. All components together allow for composing complex teaching applications from media objects and streaming them to the Web.

This paper is organized as follows. In section 2 we introduce the basic ideas of our Media Object Model and exemplarily compare to related works. Section 3 presents the underlying multimedia database system and introduces to its authoring toolset. Principles, architecture and implementation properties of the MobIT runtime environment will be discussed in section 4. A brief introduction to further applications is given in section 5. Finally, section 6 is dedicated to conclusions and an outlook on the ongoing work.

## 1 The Media Object Model and related Works

### 1.1 Enabling active Document Structures

The teaching and presentation environment introduced here aims at the one hand at profoundly supporting arbitrary media data including time-based material. On the other hand we want to provide a flexible mechanism for structuring documents which not only accounts for thematic interrelations (e.g. links), but also gives rise to object compositions including temporal aspects or complex interaction events. Thus led by our object model and guided by its web authoring toolset an author should be able to produce for instance a multimedia information stream being the composite of any heterogeneous collection of media data (text blocks, audio, images, video, ...).

Central to our object oriented information model therefore is the notion of active references as a basic composition mechanism. These interrelations not only carry the ability to refer to subordinate presentation data, but are capable of imposing event-type actions on its references. As typical notions of referential actions between documents we consider the connections in time and (presentation-) space or the predefinition of possible user interactions. Since these active references are foreseen in the data object model authors may

by a simple editing of attributes build up a document structure which inherits some active processing from its information object class. The individual mark-up of active document structuring is defined with the design of an educational application. Structuring process has been kept very flexible to permit application-oriented, semantically meanings, thereby donating intuition to authors when dealing with the system.

As pointed out above an important role is dedicated to the reusability of document data. As it is of course easy to assure multiple exploitability of simple media files, reusability of complex, composite media documents is of much higher importance. These collections of interrelated documents usually play the role of knowledge modules and bury significant amounts of authoring work. They are also subject to singular change, depending on the knowledge evolution. Our model does support for reusable presentation components by providing a uniform, media independent data structure which we denote by Media Objects. Mobs serve as universal containers for embedding either subsequent Mobs or media data. By referencing one another Mobs allow for arbitrary compositions of unlimited complexity, where the atomic nodes of the resulting graph structure are formed by distinct Data Objects. Besides its uniform appearance the Media Object entities support for application reuse in restricting active links to referenced objects thereby relying on referential integrity ensured by the underlying database system.

## 1.2 Media Objects

Media Objects may be seen as the central constituents to comprise the data structure of our object model. As the basic design idea a Mob consists of both, the subordinate reference list and the collection of active references, the latter being restricted to act on Mobs included in the reference list. Simultaneously arbitrary annotations may stick to these hulls, neutral with respect to applications or actual media data.

Defining an application at first requests for turning the Mob structure into a meaningful formation. Semantics can be brought onto Mobs in a twofold fashion: Dynamic content processing may ground on (mandatory) attributes assigned to the data and thereby organize content according to meta information. Media objects in this first step remain singular informational units. The powerful approach however lies in interpreting the active references native to Mobs. An application designer not only can choose from arbitrary interrelations such as trees, Petri-nets, circuits, . . . , but can dedicate operative instructions to those data links ranging from a simple automated Web-link generation over spatial and temporal construction policies up to conditional interactions within arbitrary scenes.

## 1.3 Related Works

Enumerous activities rank around document structuring and authoring of more complex information models than HTML-formatting. From the educational area we exemplarily mention the group of Maurer [3],[4], who propose and implemented the Hypermedia Composite Model as a semantic container for learning documents. Even richer research is going on in the area of multimedia database systems. For an excellent overview we refer to [6].

As mentioned earlier several interesting research activities are presently enforcing the notion of time to the Web, the most prominent being the W3C recommendation Synchronized

Multimedia Integration Language (SMIL) [10]. As a declarative language SMIL allows for synchronization of media objects in an somewhat simplistic, HTML-style fashion. Synchronization is done in object pairs, either sequential or in parallel. The appearance of any object may be bound to a duration parameter. SMIL extends the meaning of hyperlink to connecting temporal and spatial subregions.

The runtime behavior of any SMIL interpreter thereby is more or less left open, which probably is the most important drawback of the model. Combined with the absence of a stringent handling of timelines temporal inconsistencies in more complex documents can be foreseen. Besides few reference implementations of SMIL players there is an attempt to include synchronization features into the Web browser named HTML+TIME [11]. This proposal addresses temporal extensions to HTML and incorporates basic elements of SMIL.

Both ideas however suffer from strong limitations due to the simplistic ansatz of HTML omitting any structuring for media object use. Rutledge et al [12] consequently report about severe difficulties in authoring SMIL presentations mainly due to the lack of reusability for object compositions as well as SMIL's inability to deal with complex object relations. In most recent works, the 2.0 specification of SMIL [10], the World Wide Web Consortium heads towards a realization of SMIL as a module within the framework of the XHTML language. Most of this work is presently ongoing and far too incomplete from permitting implementations.

As a completely other example more similar to our work we like to mention the Nested Context Model (NCM) of Soares et. al. [13]. With the aim of grounding a strong structure for flexible deployment of hypermedia documents the NCM provides a composite meta-structuring for different media types, called nodes, up to an arbitrary level of complexity. Those nodes may contain a reference list of denoted nodes giving rise to an arbitrary graph structure of the composed document. The model, which has been implemented in a system called HyperProp, treats hypermedia documents essentially as passive data structures. Synchronizations define through events which may occur as the result of object presentation or user interaction.

Since embedding of media objects within the NCM results in a passive mesh without further presentational meaning, an additional structure of activation, events and contexts (called perspectives), has to be superimposed. This characteristic on one hand leaves some liberty to the author (the same object structure may encounter different behavior in different contexts), on the other hand it adds an additional level of complexity to the modeled hypermedia system and denotes the major difference to our work.

## 2 MIR - A Media Information Repository

### 2.1 The Media Object Database System

The core of our multimedia environment is formed by a media object database system. Named Media Information Repository (MIR) it combines all operations related to data storage and at the same time keeps track of information structuring ensuring referential integrity. Although MIR fully implements the media object model it remains neutral with

respect to applications built on top of the database layer. The intention in designing the media repository was to provide a robust, powerful basis, on which a multitude of educational systems may be established with rather limited effort.

MIR divides into two functional groups: The Media Object Lattice and the Data Store (s. fig 1). Objects in both repositories may be addressed by symbolic names embedded in a virtual file system. Besides administrative information concerning owner, group and access rights data entities can carry arbitrary annotations by means of an open property list. Technically only distinct by data type definitions properties may contain any kind of meta information, e. g. content descriptors such as subjects and keywords, didactic annotators concerning presentation order or information depth, and technical markers being specific to the educational applications on top of the database, as well.

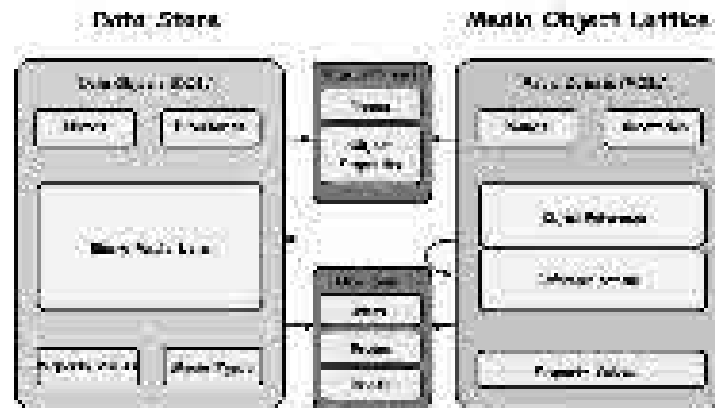


Figure 1: Media Object Database Architecture

Media entities in our object oriented database design belong to classes, which define their properties. Any object instantiates the class its derived from and thereby inherits the property set including type and attribute definitions. Customisation of the MIR to support a new application thus limits to the set-up of appropriate object classes with the possible need for extending authoring functions (see below). Quite independent of actual exploitation the universal data structure MOB is offered for application processing.

As an advantage of the MIR data logic actual media handling separates completely from application design: The Data Objects (DOBs) reside in a Data Store together with its media descriptors, rsp. mime-types. They are ready for multiple access by either MOB-based applications or directly through http-requests, where media specific treatment has to be taken care of by clients and - if necessary - by middleware components independent of the actual application.

## 2.2 The MIR Authoring Environment

Easy access for authors the system grants through a Web authoring tool. It is designed to guide through the different levels of complexity by means of several adapted views. As it is well known and to some extent obvious that the WYSIWYG paradigm does not hold in the case of temporal, structural or event editing [5], we attempt to relate the multiple aspects of authoring to specific, intuitive appearances of the tool, thereby relying on the semantics of structural relations mentioned above. Application design by means of an object class editor though carries no presentational meaning and remains rather formal as its use might be restricted to technical staff.

At the first stage of content authoring our tool allows for DOB upload and control. Guided by an object browser the author may organize and retrieve objects in a directory structure of a virtual file system, donate names, media types and properties to the DOBs and upload actual data to the MIR data store (s. fig. 2). In general media data manipulation is not meant to be part of the application authoring process, but for the sake of simplicity a simple text editor which also supports for HTML-formatting is included in the system and permits the direct generation of written text.

Whereas the object browser in the MOB editing regime remains unchanged, dedicated support is given to the author in designing presentations. With the help of a structure view, a spatial view and an (relative) object timeline authoring of MOB-based applications receives its basic instruments. As was pointed out above, however, the specific semantic of media object structures is only fixed with application layout. The authoring requirements thus may significantly change between different fields of use and specific aspects cannot be foreseen in general. Including a toolbox of methods our open system therefore provides a programming interface to allow for easy, application dependent extensions in the form of specific views.

## 2.3 Architecture and Implementation

The technical concept of the MIR environment is formed by an open multimedia architecture designed according to a 3-tiered principle as is shown in fig. 3. Implementation thereby followed the major goals

- **Functionality:** The environment - meant as a uniform platform for media object processing - must provide all fundamental operations on MOBs and DOBs like load, store, search etc.
- **Flexible Media Handling:** The system must adapt to a general range of possible media types including discrete and continuous objects.
- **Standard Conformance:** All used or implemented components should conform to present standards established. User or application interfaces should rely as far as possible on application standards s.a. the Web protocol, mime-type handling, streaming protocol standards etc.
- **Performance:** According to well known resource requirements of (continuous) media data specific measures concerning leanness, optimization and scalability of the system should be applied.

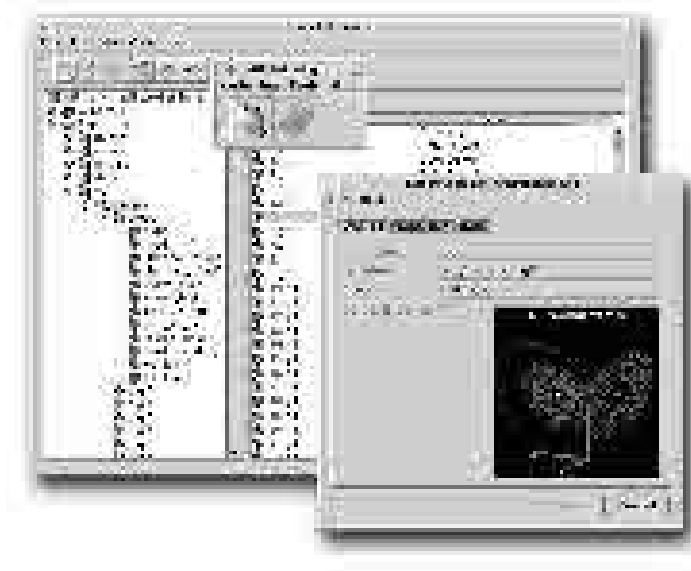


Figure 2: The MIR authoring tool

- **Encapsulation:** Access to media object data should only be granted by a set of appropriate, general operations, thereby hiding low level manipulations such as SQL-statements.
- **Extensibility:** Characteristics of applications as well as additional media types can be expected to impose specific requirements to the environment. Besides a uniformly suitable data environment application and media processing units need to offer universal programming interfaces for adding the requested capabilities to the system.

The current implementation of the media object database runs in a relational database management engine, a Sybase adaptive server, with special tuning applied to it. This platform we chose as a robust, very fast and lean basis. For the sake of encapsulation and performance, but also to achieve an ‘object oriented’ data modelling all data accesses and manipulations are realised by means of stored procedures. Media specific operations such as compression/decompression, streaming or synchronisation tasks are performed by the middleware components, since middleware services are scalable, support load balancing and in our case accommodate caching.

All middleware components are written in JAVA and are primarily responsible for the session and transaction management and for a buffering cache layer which allows for latency hiding. Even though we employ a single component server solution, the Sybase Jaguar, most of the implementations fulfil the JAVA EJB specification and are therefore rather neutral with respect to the specific product. Client access is granted in a manifold way (s. fig. 3): On the standard side the natural IIOP-exchange of objects is offered to intelligent client apps complemented by standard Web protocol http for all public entities in the database, the latter being implemented by a servlet in the back of an apache/tomcat

installation. As serialisation of binary large objects forms an inefficient way of transport we decided to incorporate the Sybase proprietary transaction protocol TDS, which shuffles binary data in bulk. The client programming interface for TDS-transport is hidden behind JDBC, so that proprietary code can be kept from application programming.

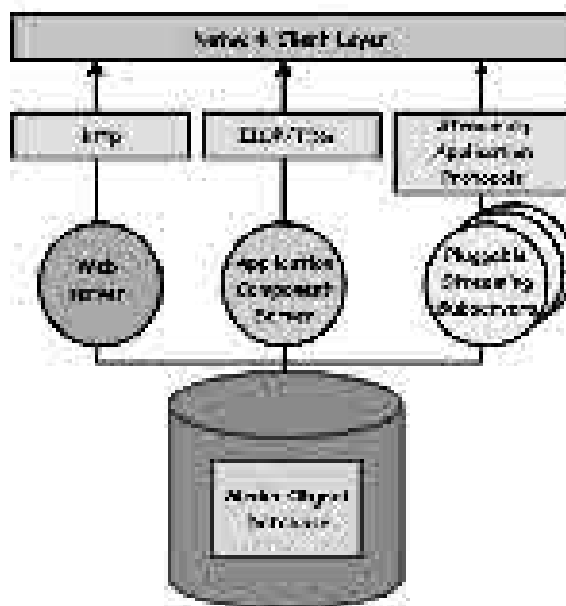


Figure 3: Networked Architecture

As an important feature of the platform introduced here may be seen its ability to deal with pluggable subserver (s. fig. 4). Subservicing not only opens up the field for application dependent media streams, but also allows for incorporation of new, complex functionality such as online data processing without fattening a thin applet client. For an overall stream oriented system it appears quite natural to include served media for streaming and such. MIR provides a flexible and simple interface for this purpose. In current applications subserver are used to incorporate the high performance optimized JAVA Wavelet video player of Cycon et al [7], a direct text sender which permits messaging to ongoing presentations and an MPEG3 server which processes audio.

The interface to include any type of subserver has been purposely designed in minimal fashion: Any subserver in perspective must implement the methods `getPortCount` to allow for inquiry on requested number of ports, `setPorts` to permit port assignment, `setData` to receive data handles and the initialization. Additional information classes etc. are kept

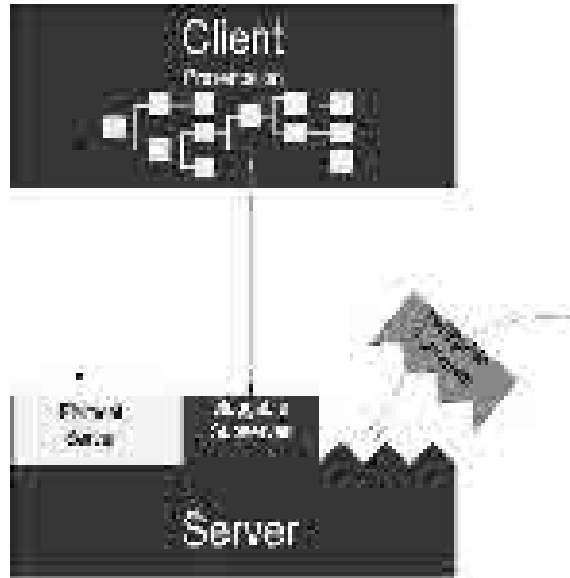


Figure 4: Subserving as a Plug-In

optional. The interface at the corresponding client site appeals as even simpler: `setServerInfo` and `getServerInfo` are the methods needed here. Within this open framework it should be easy to bring additional data servers to the system as for instance to include real-time visualization or live streams or ...

Implementations on the client side merely depend on application complexity than on guidelines taken from MIR environment: Clients may be applets based in Swing like our authoring tool, simple HTML-pages or Servlets running JAVA Beans in correspondence to JAVA Server Pages. Any time-based application we however have not undertaken without browser's JAVA machine.

### 3 Media Objects in Time

#### 3.1 Presenting a media stream

As one major application the teaching and presentation system Media Objects in Time (MobIT) centres about the idea of media objects synchronizable in time which may be linked to form fairly complex presentations. But at the same time any object remains self consistent and of independent use. Roughly speaking our basic concept consists of defining media object instances and lining them up in time as is shown in figure 5. MobIt intends to provide an accurate scheme for temporal and spatial placement of presentation objects, where authors neither have to take care of interobject synchronization dependencies nor adaptation to possibly inaccurate network performance, the latter being subject to implementation of latency hiding techniques.

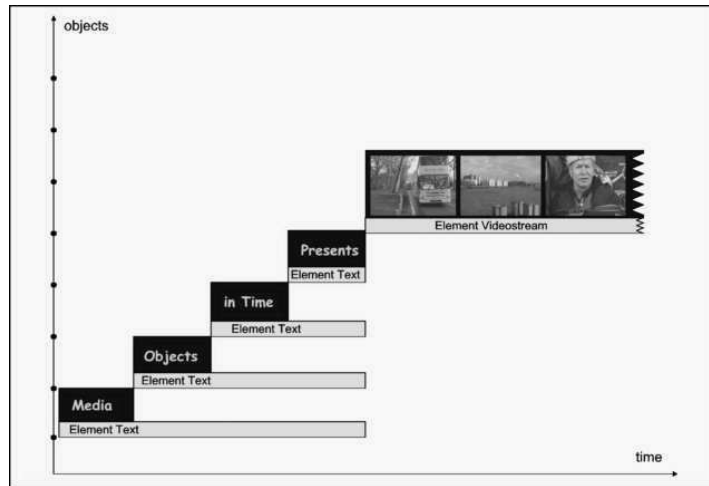


Figure 5: Media Object Instances in Time

Presenting itself on a timeline any presentation becomes a time-based data object, even if composed only from timeless media such as texts or images. Any presentation component will carry an instance of initial appearance and a moment (possibly at infinitum) for fading away from the client's screen. Within this framework of MIR any streaming media such as video or audio may be included and synchronized to the scene and the overall data stream.

Aiming at the combined utilisation in lecture rooms as well as teleteaching our model focuses on a clear, straight forward concept of reusable compound media components. Any of these will be accompanied by screenplay scripts arranging their behaviour in space and time. Thus in place of the page oriented WWW concept or the typically event driven nature of CBT products MobIT runs as a flow oriented presentation model showing for example a crash-test video combined with charts of relevant statistics and vocally explained CAD car models in subsequence.

### 3.2 The Compound Flow Model

In designing an educational system within our environment structuring has to be given an applicational meaning. In the context of MobIT this is done by the Compound Flow Model (CFM), which takes much care to define a simple structure of straight forward logic intuitively appealing to document authors. The CFM organises the uniform hull entity Mob in a tree structure, where any branch reference expresses a temporal and spatial inclusion relation. (s. figure 6).

Media objects form the central construction element of the CFM data structure. As bound to the basic design idea of MIR Mobs include the subordinate object reference list and a screenplay script acting on the references, thereby describing all parameters responsible for their behaviour in time and space. Those scripts we denote as Playlists. Playlists describe the states attained by the corresponding Mobs in total.

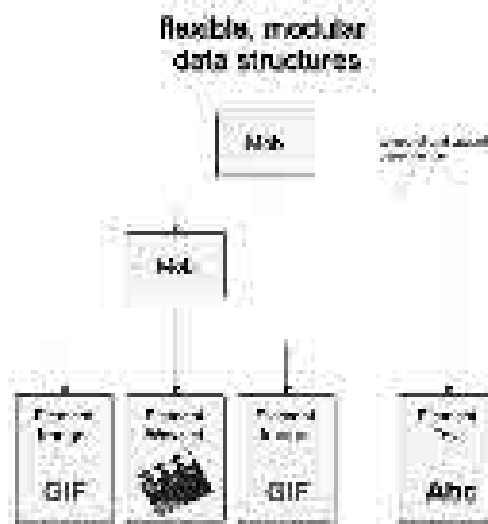


Figure 6: Media Object Hierarchy

Tightly bound to the concept of combined reference to objects and their states is the notion of generalized reusability for any component involved. Roughly speaking an object exhibits generalized reusability, iff it is self consistent, i.e. free of recursions, and parametrizable in state space. The fundamental parameters of the state space up until now are the spatial size and the duration in time. Some additional features such as background color or font type change have been implemented.

Vital to the framework of CFM is an environment for generating and controlling the flow. As media objects for a given presentation may be widely branched, each one of them equipped with a complex structural inheritance and its own synchronization demands, a flow control module needs to resolve all structural data dependencies. It thereafter has to linearize resulting bulk information, to form an ordered flow and at last addict objects to the externally provided primary timer.

Even though components of the Model are of active, self consistent nature an additional flow generator needs to be present. Generating a flow in our context has to fulfil the task of resolving all open object dependencies, collecting the data and en passant performing co-ordinate transforms and at the core linearize data with respect to time. As a result of such linear alignment all playlists are merged to form a complete script for the screenplay the whole presentation consists of. Additionally may be observed that the flow generator as described is - if properly implemented - well suitable for transmitting presentations data collection as a sequential stream over the network. For a more detailed description of the MobIT application see [2].

## 4 Further Applications

### 4.1 Virtual Design

The design studio of tomorrow will not contain a computer anymore, but will consist of the computer network. Guided by this maxim a completely different idea of computer based educational system has been developed in collaboration with Bildo Akademie für Kunst und Medien Berlin. Interactive picture networking has been adopted as a basic co-operative internet platform for designers of digital images [8],[9]. The project has been honoured in the meanwhile with the "New Talents Award" at the direct marketing congress DIMA in Düsseldorf 1998 and the special price Multimedia Transfer at Learntec 2000.

People from art and design communicate through their visual products. As it is rather difficult to circumscribe representational and aesthetic contents in standard language terms, a specific way of expression needs to be utilized: A Language of Pictures. Like any stream of statements such a visual speech needs basic order principals, a timeline and thematic assignments at minimal.

The Virtual Design project started from the idea of supplying a networked communication platform which allows for creation of visual dialogs. Starting from a "white canvas" each participant is enabled to contribute data sets consisting of an image, a title and a textual commentary to the system. The system itself requests such contributions to be a reaction of a former entry. It thereby links entities and lines pictures in time chains, optionally branching at nodes which invoked multiple reactions. As time evolves the Virtual Design system will give rise to a tree of pictures with each branch representing a visual dialog between authors (s fig. 7).

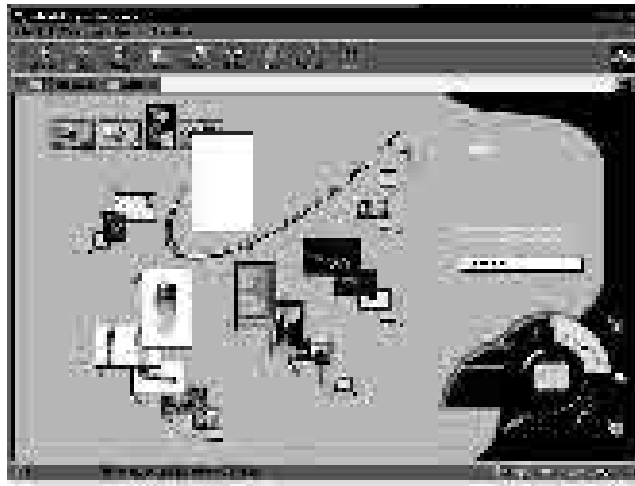


Figure 7: Virtual Design Visual Navigator

Relying on MIR basic environment Virtual Design MOBs enclose images, thumbnails and textual complements. The media object structure in the virtual design application is defined by the virtual dialogs performed by using the system and is assigned automatically as part of the work process. Note that no separate authoring is needed since VD combines workspace and presentation.

#### 4.2 Knowledge Café

As a third, much simpler application of the MIR data environment we want to introduce a small knowledge café prototype. The system ranks around pieces of information which are classified according to topics and keywords and with respect to information complexity, as well. Generating content-based meshes from Mob references the information repository not only is able to answer property related searches but will dynamically present document groups as Mob references are automatically transformed into Web links. With the use of JSP-techniques this useful application could be developed in a very limited number of days by relying on the strength of the MIR technology basis.

### 5 Conclusions and outlook

The multimedia information technology presented in this paper is an ongoing project in many ways: Having accomplished an efficient basic solution on structured media processing several teaching applications to be used either in the lecture hall or at students homes are to be implemented. Most exciting however we consider future developments in the area of time-based learning and presentation system.

Much work however has to be done in this ongoing project. Interactions have not been defined in CFM yet. As simple smil-type hyperlinks in our flow oriented model could only support hopping between - possibly nested - parallel timelines and as we do not intend to produce some sort of interaction programming language, our current activities concentrate on modeling an interaction paradigm. Accounting for the CFM potential to operate on self consistent media objects we are aiming at a small 'alphabet' of operations which enables authors to open up an unlimited number of navigational paths to the receptor with only a limited number of interactions defined.

Interactions will introduce an additional complexity to the treatment of network behavior as they might contradict latency hiding techniques in some parts. This is unavoidably true for user dialog elements. For the loading of binary large elements a careful time control will be needed. The buffered pre-reading of Mobs however may be viewed as filling an instruction cache of a processing unit. Interactions impose branches to the instruction flow and can be buffered in parallel so that immediate system response generalizes.

#### Acknowledgements

We would like to thank Hans Cycon and his group for enjoyable co-operation. They developed the highly optimized Wavelet Video algorithms and the codec. Mark Palkow carefully ported the codec to JAVA.

## References

- [1] Jourdan, M., Layaida, N., Cécile, R.: Authoring Techniques for Temporal Scenarios of Multimedia Documents. Ed.: Borko Furht: Handbook of Internet and Multimedia Systems and Applications. Boca Raton, Florida, CRC Press LLC, 1999Page179 - 200,
- [2] Helic, D., Maurer, H., Scherbakov, N.: Authoring and Maintaining of Educational Applications on the Web. Proc. ED-MEDIA'99. 1999, Seattle, USA, AACE, Page1792-1797,
- [3] Helic, D., Maglajlic, S., Schaerbakov, N.: The HC-Data Model: A New Semantic Hypermedia Data Model. Proc. WebNet'99. 1999, Charlottesville, VA, USA, AACE, Page493 - 498,
- [4] Aberer, K., Timm, H., Neuhold, E.J.: Multimedia Database Management Systems. Ed.: Borko Furht: Boca Raton, Florida, CRC Press LLC, 1999Page285 -310,
- [5] Feustel, B., Schmidt, T.C., Marpe, D., Palkow, M., Cycon, H.L.: Compound Media Streaming in Time. Ed.: Vaclav Scala: Proc. 9-th Int. Conf. Comp. Graph., Visual and Comp. Vision WSCG'2001. 2001, Plzen, 2001
- [6] Marpe, D., Cycon, H.L.: Very Low Bit-Rate Video Coding Using Wavelet-Based Techniques. IEEE Trans. in Circ. and Sys. for Video Techn.. Vol.9. 1999, 1, Page85-94,
- [7] Karpati, A., Löser, A., Schmidt, T.C.: Interactive Picture Network. IEE Electronics & Communications. Vol.99. 1999, 109, London, 1999Page18/1-18/8,
- [8] Born, T., Hannecke, T., Heine, A., Karpati, A. Kemnitz, T., Löser, A., Schmidt, T.C.: Virtual Design Update. Vol.27-99. Berlin, FHTW-Transfer, 1999
- [9] Maurer, H.: Can WWW be succesful. IFIP - DS8. Vol.Jan. 99. 1999, 1999
- [10] Ayars, J. et. al. (ed.): Synchronized Multimedia Integration Language (SMIL 2.0). World Wide Web Consortium Working Draft. Sept. 2000, <http://www.w3.org/TR/smil20>.
- [11] Schmitz, P., Yu, J. Santangeli, P: Timed Interactive Multimedia Extensions for HTML (HTML+TIME). Note for Discussion submitted to the World Wide Web Consortium. September 1998, <http://www.w3.org/TR/NOTE-HTMLplusTIME>.
- [12] Rutledge, L, Hardma, L, v. Ossenbruggen, J: The Use of Smil: Multimedia Research currently applied on a Global Scale. Ed.: Karmouch, A: Proc. of Multimedia Modelling 1999. 1999, Ottawa, Canada, Wolrd Scientific, Page1-17,
- [13] Soares, LFG., Casanova, MA., Rodriguez, NLR.: Nested Composite Nodes and Version Control in an Open Hypermedia System. Intern. Journ. on Information Systems. Vol.20. 1995, 6, Elsevier, 1995Page501-519,