TIEMPO: An Authoring and Presentation System for Interactive Multimedia

Stefan Wirag, Thomas Wahl, Kurt Rothermel

Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)
Breitwiesenstraße 20-22
D-70565 Stuttgart
e-mail: wirag@informatik.uni-stuttgart.de

Abstract

The composition of interactive multimedia titles is a challenge for all advanced multimedia authoring tools. When non-interactive titles were specified by aligning the media items on a temporal axis such as time lines or tracks, now interactive multimedia titles require a non-linear temporal model. Thus, authoring becomes more complex, and the structure of a title is more difficult to visualize. TIEMPO[1] *addresses two areas: First, a temporal model satisfying the specification needs of interactive multimedia; Secondly, an interface for easy and rapid authoring of multimedia titles. This paper presents the architecture and design principles of the* TIEMPO *authoring and presentation environment.*

1. Introduction

An interactive multimedia title is a composition of media items, interaction elements, temporal and spacial interrelations and other presentation information. When composing a multimedia title, the author selects the media items, which can be of any media type such as text, video, audio, graphics etc. Then, the layout of the multimedia title is defined by arranging the media items in time and space. Finally, interaction elements and further presentation information such as volume, colors etc. are added to the title. So, authoring of a multimedia title is defining the structure of interrelated media items.

Usually in a distributed environment, the structural description of the multimedia title is stored apart from the media contents. Also, the structural description itself might be distributed applying referencing techniques in the composition of titles. Figure 1 shows how a distributed multimedia title is presented. The structural description of a multimedia title is processed on a presentation terminal, that is connected to a network, on which a number of media servers are

1. *TIEMPO*: grant of the Deutsche Forschungsgemeinschaft DFG
Temporal **i**nt**e**grated **m**odel to **p**resent multimedia-**o**bjects

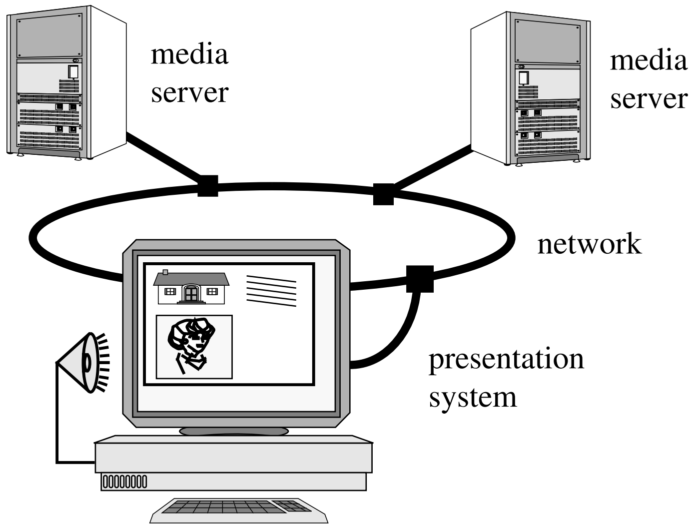


Figure 1: Presentation scenario

available. When rendering the distributed title, the media items and structural descriptions of presentation parts are transported over the network to the presentation location.

To guarantee a high-quality presentation over the network, synchronization mechanisms are necessary to render the document as intended by the author. Synchronization is implemented on two levels: The scheduling and the stream synchronization level. Synchronization on the scheduling level assures that the media items are started, stopped and manipulated on time. Therefore, it applies coarse-grained synchronization mechanisms. *Tiempo* provides algorithms to map the temporal specification of a multimedia title to a runable schedule.

On the stream level, fine-grained synchronization mechanisms are implemented to guarantee the synchronization of the data streams, e.g. on the level of lip synchronization. *Tiempo* uses the multimedia platform *Cinema* [BDH+93] that provides stream synchronization mechanisms. A survey on stream synchronization mechanisms is given in [EFI94].

The general goal of the *Tiempo* project is to provide a temporal model that enables the author to specify scheduling and stream synchronization constraints for non-linear, interactive multimedia. Adding interaction to multimedia titles makes their temporal layout quite complex because the user might interfere with the presentation at any time. Thus, the temporal layout of a multimedia title varies for different executions of the title depending on the instants and forms of user interaction. To cope with this complexity, a second goal of *Tiempo* is to offer an authoring interface for easy and rapid composition of multimedia titles.

This paper presents the system design and architecture of the *Tiempo* authoring and presentation environment. Further information on temporal modeling can be found in [WaRo94] and on interaction and user interface modeling in [WWR95].

The paper is organized as follows. In section 2, we describe the research goals of *Tiempo*. In section 3 we describe the applied notations to describe specifications for various purposes. Then, the architecture of the presentation system is presented. In section 5, we take a glance at the prototype that we have developed. In section 6, we compare existing approaches and analyze document standards. Finally, we summarize the main issues of *Tiempo* and present our current field of work.

2. Goals of *Tiempo*

The goals of *Tiempo* can be divided in three correlated categories: temporal model, specification representation and tools. The development of the temporal model is the fundamental part of the project. The goal of the second category is the development of a multimedia document language allowing the specification of temporal relations at a high abstraction level. Further, a presentation system is built applying the language and including an authoring tool, a presentation tool and various translators and formatters.

The *Tiempo* project provides a fundamental temporal model in the context of distributed multimedia documents. The following issues are covered by the temporal model:

- *multimedia complete*

  All relevant temporal relations that occur in multimedia documents should be representable within the model, i.e. ‘A temporal model is multimedia complete’ indicates that all necessary temporal relations are included in the model.

- *quantitative indeterminism* [KeLo91]

  Temporal extents which are necessary to perform a presentation can be indefinite at authoring level. Thus, it must be possible to model temporal durations in an indeterministic manner. The temporal model must integrate indeterminism to ensure the flexibility of created documents.

- *qualitative indeterminism* [KeLo91]

  There might be presentation parts whose interrelations are not definite at authoring time. Thus, it should be possible to specify partial restrictions of the relations and leave the selection of the best relation to the system.

- *interaction*

  Interaction is a crucial topic in enhanced multimedia presentations. Especially, interaction with a temporal impact have not been analyzed satisfactorily. Thus, the temporal model should not only represent temporal relations but also integrate temporal interaction. Further, it should be possible to model synchronous and asynchronous interaction. With synchronous interaction, the system waits for user input. Asynchronous interaction occurs when the user intervenes in the running presentation. Thus, asynchronous interaction is more complicated to handle.

- *explizit and implicit relations*

  A specification must be unambiguous. For any specified relation, the system has to known whether it is mandatory or optional. Usually, mandatory relations have to be specified explicitly. Optional relations give the presentation system the choice to meet the relation or realize a different relation. Therefore, the presentation system is more adaptable to other constraints, e.g. resource limitations.

In a second step, a specification language is developed based on the temporal model. The language should meet two goals: First, it should be easy to use for non-professional authors, and secondly, it should be portable to other platforms. To achieve the second goal, it is necessary to have a formal representation. To guarantee platform independency, a standard notation might be applied such as HyTime [HyTi92] or MHEG [MHEG94].

Finally, to test and evaluate the developed model and language, an authoring and a presentation tool is developed. The authoring tool generates multimedia titles in the *Tiempo* language. Authoring is supported by various system functions making the authoring process as simple as possible. The presentation tool performs the optimized presentation of a document by taking advantage of indeterminism contained in the document. Additionally, a translator is developed mapping a document from the *Tiempo* language to a standardized multimedia document language and vice versa.

3. Specification notations

Any multimedia document system must integrate document notations to handle presentation specifications. Theoretically, only one notation is needed to describe a multimedia presentation. But the authoring, exchange and presentation of documents demands notations with varying characteristics. Thus, we use different document notations in *Tiempo*. Figure 2 shows the correlation between the system moduls and the presentation descriptions.

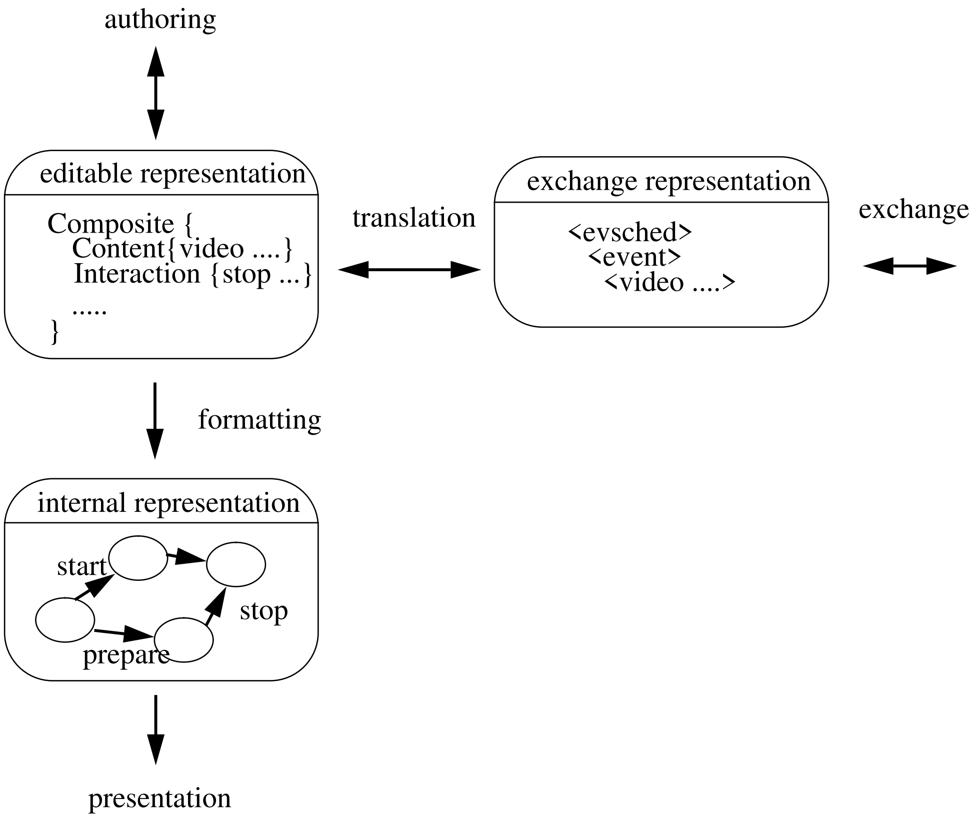


Figure 2: Specification notations

3.1 Editable notation

One goal of *Tiempo* is the development of a multimedia document language integrating interaction and flexible synchronization constraints. Multimedia document specifications in the editable *Tiempo* language are created and modified with the graphical *Tiempo* editor. Currently, we develop the *Tiempo* language providing the following features:

- A modular structure allows to store the structural description of presentation parts at different locations in a distributed system. Thus, a multimedia document can be a web of presentation specifications. For the reusability of specifications, any structural description of a presentation part can be embedded in several presentations by referencing techniques. It is possible to create documents that are composed from presentation parts in different multimedia document notations if the appropriate translators are available.
- Two major techniques are provided to combine rendering instructions and media data. First, it is possible to integrate media data that can be defined without supporting tools such as short texts or headings in the multimedia document together with rendering information. Secondly, media data can be addressed by referencing without integrating them in the rendition information.
- Media items may not be presented in the way as they were originally stored. For example, an author wants to present only the last five minutes of a video, and he wants to modify the colors in the video. Therefore, specification mechanisms to identify particular parts of media items and to describe their modification are provided.
- An abstract set of operators is used for specifying the spacial or temporal arrangement of the media items. Interaction requires flexible spacial and temporal relations. For example, the interactive speed-up of a video changes its presentation duration. Thus, synchronization relations have to be specified taking into account the possibility of interaction.
- Interaction itself is a relation type which associates a sensitive region such as a button area with the affected objects. Theoretically, any element on the screen (e.g. determined by a pattern matching algorithm) might be associated with an interaction. Thus, it is almost impossible to provide all imaginable interaction forms which might occur in multimedia documents. Therefore, we provide a set of basic interaction types and a mechanism to combine basic interaction types to complex interaction forms.
- For easy authoring, a document must have a transparent structure. Thus, correlated media items have to be grouped. The grouping mechanism allows to map hierarchical structures known from textual documents such as chapters, sections and pages.
- For rapid authoring, reusable, parameterized specifications can be defined. We integrate a generic template mechanism which allows to define arbitrary complex specifications as template. When applying templates, the parameters are associated with real media items. Thus, it is possible to define a general layout for the documents of an organization or a general panel to interactively control the presentation of videos.

3.2 Notation for document exchange

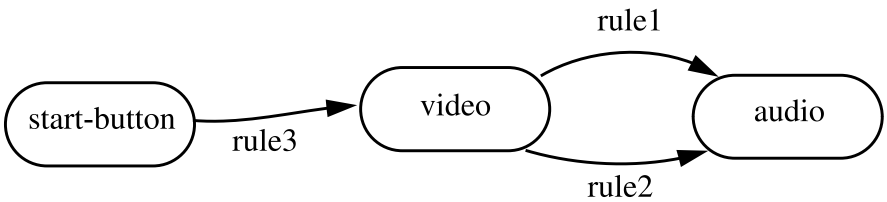
To exchange documents with other platforms and vice-versa, a standardized presentation representation is needed. An adequate standardized language has to cover the concepts of the *Tiempo*-language. A high abstraction level is crucial for any applicable standardized language.

We have analyzed specifications in HyTime [HyTi92] and MHEG [MHEG94] standards in regard to the flexible synchronization constraints and the complexity of the temporal interaction forms. But the flexibility of the features defined in those languages revealed to be insufficient although some forms of flexibility and interaction can be modelled using extensions such as integrating scripts or creating derived elements. [WRW95] suggested to extent HyTime to include various interaction forms. However, a transformation might not be done without any loss because at most those features which are conceptually predefined in the standardized language are known to all rendering machines. This means that some indeterminism types of the *Tiempo* language have to be removed during a translation to HyTime. Also, not all temporal interaction types, e.g. reverse interaction, can be modelled with MHEG.

3.3 Internal representation

To render a multimedia document, a schedule has to be derived from the presentation description. A schedule denotes for each media item when its rendition begins and ends or generally when a particular action has to be performed. Whether an explicit schedule representation is needed or not depends on the complexity of the presentation notation which is determined by the applied specification language. Low level specification languages such as MHEG do not need a transformation before the presentation because they implicitly contain the presentation schedule. But the high-level constructs of the *Tiempo* language require a transformation in a notation which integrates a presentation schedule.

Interaction and reloading of document parts during the rendition are events that require the modification of the schedules. Thus, we do not create an explicit schedule but rather use a self organizing internal representation that allows to add new specifications derived from reloaded presentation parts. The notation is based on rules describing object modifications in dependency of state changes or other conditions. Figure 3 shows the internal representation of an interactive presentation.



rule1: if (start-button is pressed) then start video
rule2: if (video is started) then start audio
rule3: if (video is stopped) then stop audio

Figure 3: Rule based schedule description

Rule3 describes an interaction effect. This rule starts the video if the button is pressed. Rule1 and rule2 ensure that the audio is presented simultaneous to the video. The example contains two types of rules. Rule1 and rule2 represent synchronization constraints, and rule3 represents an interaction relation. Generally, various other rule types are imaginable concerning the spacial layout or the appearance of any object. Thus, the model is very flexible because the condition of each rule defines the dependency of the subsequent action from the available resources, system state or object state etc..

4. System architecture

The *Tiempo* presentation system consists mainly of two components: The authoring component and the presentation component. With the authoring component, multimedia presentations are specified, and the presentation component renders the presentation. Figure 4 gives an overview of the *Tiempo* system which is composed of various moduls each performing a particular func-

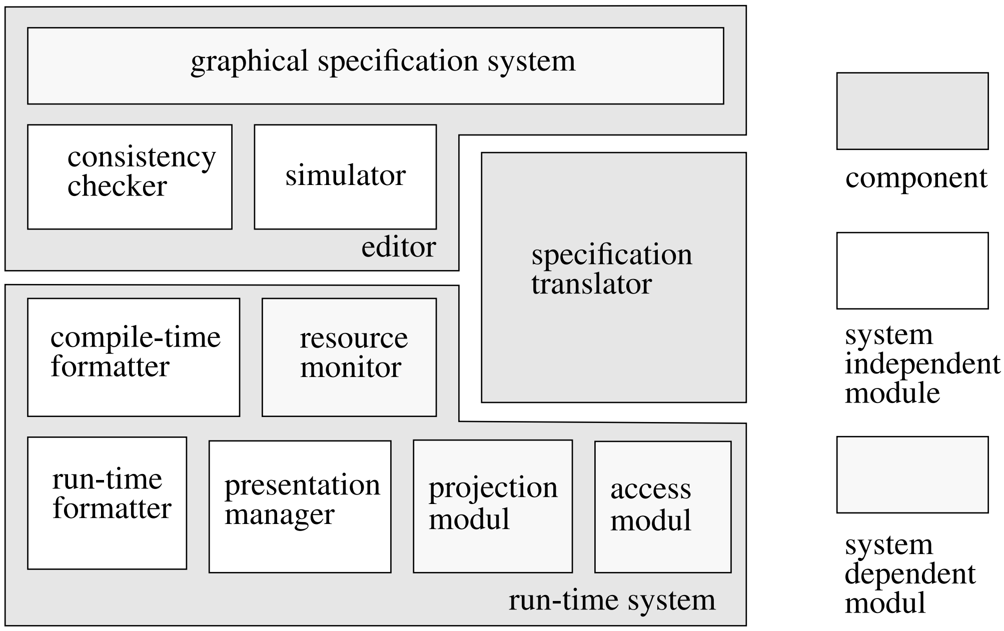


Figure 4: Tiempo-*system moduls*

tion. The modular architecture is advantageous for prototyping because it allows to replace modules in order to integrate or test different approaches without redesigning the entire system. Further, the *Tiempo* system can be easily adapted to other system environments by replacing the environment dependent modules which handle the presentation or access of media items.

4.1 Authoring component

Rapid and easy authoring of presentations is an essential goal of *Tiempo*. Thus, tools with a comfortable user interface are needed supporting the author in composing multimedia titles. A graphical document editor guarantees fast and intuitive authoring of multimedia presentations.

The *Tiempo* editor based on the *Tiempo* language provides high-level abstractions for media items, interaction and synchronization relations, i.e. the author does not have to take care of technical details of the presentation information because this is done automatically by the system. Further, mechanisms such as scratching and automatic formatting support the multimedia author. Thus, authoring is not like programming but more like composing a piece of arts. Various specification views are necessary to define the temporal and spacial layout of a document. An essential view is one to specify the temporal relations, synchronization constraints and interaction facilities. The spacial layout of media items can be specified interactively in a second view. There, specification techniques have to be applied taking into account spacial interaction such as resizing and moving objects. The system dependent interface of our prototype editor called *graphical specification system* is implemented in X/Motif [GKKZ92].

During authoring, the system does continuous *consistency checking* of the specification to guarantee a fast creation of complex and correct presentations. The result of an authoring session is a presentation specification in the *Tiempo* notation.

The authoring tool integrates a *simulator* which allows to animate a created specification or parts of it. During the simulation media items are represented by dummy elements. Thus, an author can apply try- and error-techniques to optimize the authored documents.

4.2 Translator component

The transformation of a document from the *Tiempo*-language to an interchange format and vice versa is performed by a translator component. Existing document standards tread interaction and flexibility not very extensively. Therefore, it is necessary to simplify a specification in the *Tiempo* language when transforming to a standard language. In case of a reverse transformation from the standardized language to the *Tiempo* language, we cannot take advantage of the flexible, indeterministic synchronization relations of *Tiempo*. However, the translator component integrates mapping techniques for the elements of both languages resolving the mismatch of the language capabilities. Different document standards might be integrated by replacing or extending the translator component.

4.3 Presentation component

To render a multimedia presentation, a run-time system or player is needed. Before the presentation is started, a schedule has to be built determining the temporal layout of the document. Multimedia presentations impose a high system load on computers. Thus, it is advantageous to reduce the load and computation overhead during the presentation. Thus, we integrate a *compile-time formatter* which transforms the entire *Tiempo* language document before the presentation starts. This advantage decreases if document parts are reloaded during rendition. However, we need a translation from the *Tiempo* language to the internal representation to be able to render a document. During translation, indeterminism is used to generate an optimized internal specification for the local system. A presentation is optimal if the timing requirements are met and the resource requirements are minimized. The needed information for this process is provided by a resource monitor gaining configuration information and performance indices of the local system. After the translation the internal representation integrates indeterminism in regard to interaction and resource changes by alternative rules.

A *run-time formatter* manipulates the internal representation during rendition incorporating run-time events from the presentation environment. This is done by altering the priority of rules determining if and when they are applied. A large class of events is caused by user interaction with a temporal impact. But user interaction is not the only interference with the presentation system. The underlying computer system affects the presentation as well because the given load and resources change over time. Thus, the resource monitor accumulates and maintains also dynamic resource information about the local system used to optimize preparation times with regard to synchronization constraints.

The *presentation manager* controls the rendition of the document by interpreting the internal representation. It initiates the access of presentation parts via an access modul and controls the rendition executed by various projectors of the projection modul.

The *access modul* controls the flow of media objects from the media servers to the local system. Thus, it is not independent of the local system. In our prototype, we use the *Cinema* multimedia platform [BDH+93] to deal with stream synchronization. *Cinema* provides services and abstractions for the dynamic configuration of distributed multimedia applications. *Cinema* clients may define arbitrary data flow graphs, connecting various processing elements called components. Further, it provides the concept of clock hierarchy for inter-stream and intra-stream synchronization. Thus, the access component is simple because it mainly controls appropriate services provided by *Cinema*. If *Cinema* was not available, the access component has to perform the stream synchronization. Rendition information contained in accessed specification descriptions is delivered to the compile-time component which transforms them and adds them to the internal representation. Media data such as the frames of a video are delivered directly to the projection modul. If an accessed presentation description is not in the *Tiempo* language the translator component is used for translation.

The *projection modul* renders the media data. Further, it manages user interaction such as mouse button clicks, and makes them available to the run-time formatter. It consists of a set of media projectors with different features for various media types and display techniques. The projection modul depends on the graphical interface of the local system (in our system: X/Motif).

For compile-time and run-time formatting, a synchronization strategy is used to guarantee optimal resource usage and to meet the synchronization constraints during the presentation. In contrast to synchronization of non-distributed multimedia documents, we have to cope with the problem of highly indeterministic delays during the start-phase of the presentation of media items because of the variations in the start-up times of the software components in a distributed system. Thus, the crucial problem is to determine the preparation time of a media stream such that the correlated synchronization constraints are met.

Existing synchronization schemes and architectures mostly deal with modeling of the multimedia synchronization problem without taking into account resource limitations [PrBo94]. But with limited storage facilities or tight playout deadlines (e.g. in case of user interaction) the performance of the subsidiery supporting system must also be considered. Thus, methods to adjust the preparation to the dynamically changing resources (e.g. buffers for prefetching media data) have to be applied.

The potential of synchronization strategies is theoretically not limited and might include artificial intelligence techniques to gain and apply information which helps to reduce resource requirements. For example, an improved strategy might consider user habits to optimize the preparation. If a user generally does not use help topics integrated in a presentation, their preparation can be neglected in case of limited system resources. The development of synchronization strategies is still not finished and the presentations are not optimized so far.

5. Authoring scenario

We have defined an interval based temporal model [WaRo94], which is used by the *Tiempo* language. Currently, we have just finished the first version of the graphical document editor prototype. Figure 5 shows the temporal view of the editor with a presentation specification.

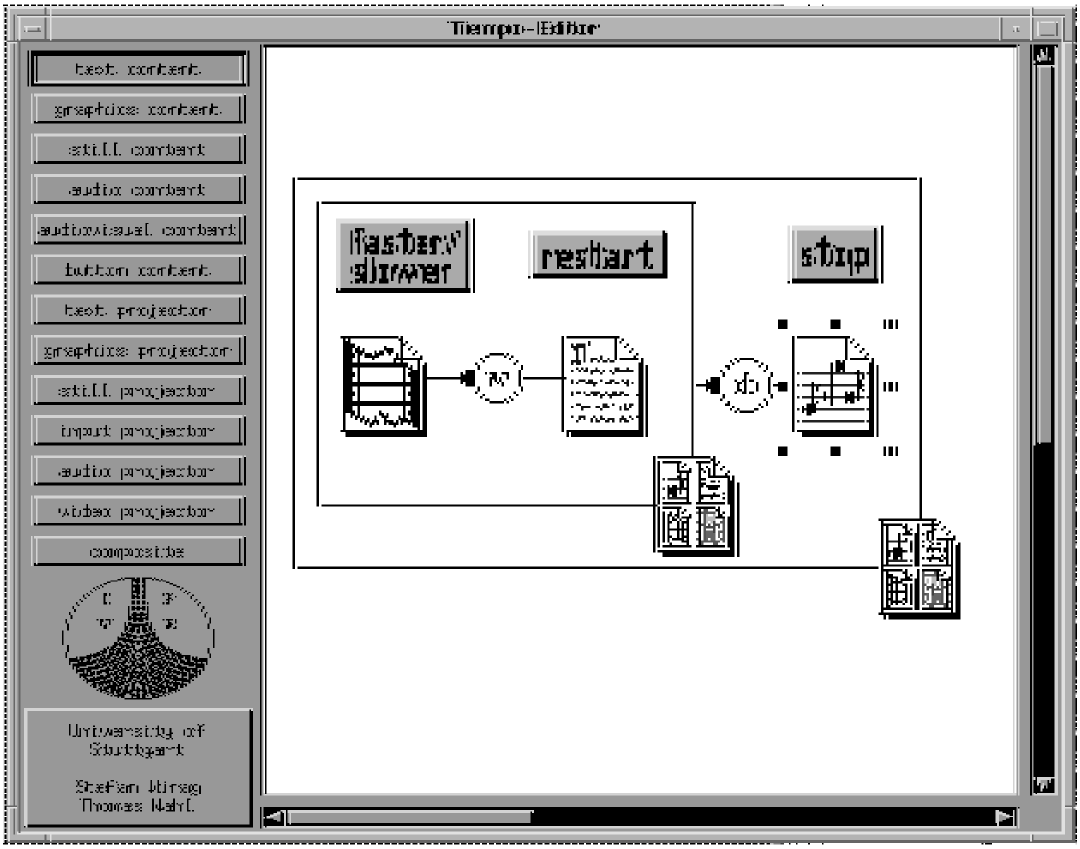


Figure 5: specification scenario

As the editor is object-oriented, the graphical elements representing media items or relations have their own pop-up menus to specify presentation details that cannot be defined interactively with a mouse. Media items are represented by icons. Relations between media items are represented by lines with a symbol denoting the relation type. The editor allows to group correlated media items. To get transparent representations on the screen, groups might be displayed with all of their contents or by an abstract icon. Interactive elements like stop-buttons appear within the group in which they can be used. Lines are used to relate interactive elements that affect multiple groups. Groups or single media items can be stored in files and reused in other specifications either by copying or by reference.

To create new elements, the author chooses the element type, and positions the element by a mouse-click in the drawing area. The selection of existing elements for manipulation, storage or presentation is done by clicking them before starting the appropriate operation.

The specification in Figure 5 shows two nested groups. The inner group contains a video with a simultaneously presented description. A faster/slower-button allows to modify the speed of the video and text, and a restart-button might be used to restart the rendition of the group. The outer group contains an audio-item with some background music and a stop-button to terminate the presentation. The rendition of the inner group should be started after the music has started. This is denoted by the relation element between the audio-icon and the group area.

Currently, the system allows to render a group or single media item by interpreting the specification without taking advantage of indeterminism or considering resource restrictions.

6. Related Work

Any multimedia authoring system needs a mechanism to store and retrieve presentation information. Former authoring systems used proprietary formats for storing their multimedia documents as no standardized languages were available. Standardized document languages guarantee the over-all exchangeability of multimedia documents. Therefore, we looked at the existing ISO standard HyTime and drafts of the standardization process of MHEG (Multimedia Hypermedia Expert Group). Further, we examined various existing authoring systems to learn from the previous design and principles of multimedia authoring.

6.1 Document standards

Various document standards (HyTime, MHEG, Script/X [Kale93], OMFI [BuGo94], PREMO [HCD+94]) have been developed to support the interchange of multimedia presentations between heterogeneous platforms. These standards involve different specification levels and differ in their capabilities in respect to indeterminism and interaction.

HyTime [HiTy92] was developed as an international standard for structured representation of hypermedia documents for integrated open hypermedia applications. It is an SGML application and is interchanged using ASN.1 for OSI compatibility. HyTime standardizes those facilities dealing with the addressing of portions of hypermedia documents and their component multimedia information objects including the linking, alignment and synchronization of document items. HyTime does not standardize the data content notation, the encoding of the information objects or the application processing on them. Compared with other standards HyTime allows authoring at a relative abstract, high-level.

The HyTime standard does not impose any particular implementation architecture, and it is possible to integrate HyTime-processing with application programs if desired. The HyTime architecture is modular and only the required facilities need to be implemented. HyTime is an encompassing standard. Thus, new element types can be derived from element types defined by HyTime. The element types can have particular semantics. Further, it is possible to integrate scripts defining application tasks.

Synchronization constraints can be defined between events which give media items a position and an extent in a coordinate system and are organized in schedules. It is not possible to define indefinite relations within HyTime nor it is possible to specify interactions directly. However, simple interactions can be specified using alternative schedules connected by links. Then, an interaction would cause the traversal of the link, and the interaction effect is generated by an adaption of the presentation schedule.

In contrast, the *MHEG* standard [MHEG94] implies authoring at a relative low abstraction level. The MHEG group began as an informal group in early 1989 to address the coded representation of a final form multimedia and hypermedia objects that will be interchanged across services and applications by storage media or networks. The MHEG standard does not impose a particular architecture of MHEG presentation systems. It only defines objects which can be used to describe a multimedia presentation. A very important MHEG object type is the composite object which is a container for content objects representing media items and objects describing the behavior of the content objects. Reflex behavior, i.e. adapting the document to current system requirements, is provided by a link-action mechanism, which also specifies the spatial, temporal and conditional relations between content objects [Pric93]. A MHEG link includes trigger conditions defining when an action object should be sent to an content object. Action objects change the presentation state of the receiving object e.g. increasing audio volume or changing size. Trigger conditions base on object state changes, e.g. from running to not running.

MHEG identifies four levels of synchronization. Close synchronization, e.g. lip-synchronization, is provided by other standards as MPEG [MPEG93], and is not defined by MHEG. Spatio-temporal synchronization is provided by positioning objects in relation to others within a coordinate system. Conditional synchronization is done by using the link-action mechanism. Other synchronization techniques can be integrated using scripts which are not defined by the MHEG standard.

MHEG defines two types of interaction objects: selection and modification. A selection allows to choose between alternatives to proceed with the presentation. A modification allows to alter content objects during the presentation. Interaction types that modify the presentation speed of objects are not supported.

In summary, all current standards have in common a rigid specification of temporal relations. Interaction is reduced to selection or modification. Most standards are extendible by integration of scripting languages. But to guarantee the exchangeability of documents, all features have to be standardized.

6.2 Authoring systems

Research has be done on some aspects of multimedia authoring and presentation. This subsection presents a summary of some predecessors of the *Tiempo* system focusing on their most advanced and exciting features.

MAEstro [Drap93] is a distributed, multimedia authoring environment. The MAEstro system includes various media editors which are responsible for the manipulation and presentation of a particular source of information, such as a video, audio or text. A particular time-line editor is responsible for structuring multimedia documents by coordinating the actions of the media editors.

To create a document, the author positions previously defined media segments (e.g. a video-clip from a video-disc) on time-lines associated with the media editors. The position and extent of a media segment on a time-line determines its relation to other media segments and its duration. MAEstro does not know any explicit synchronization relations between media segments. Thus, media segments can be aligned to eliminate start-up delays by doing a test-run. However, start-up times may vary in different runs and the alignment is not adequate. MAEstro's interaction facilities are limited to a control panel to control the whole presentation.

Firefly [BuZe92],[BuZe93] is a conceptually advanced authoring systems. It allows button-oriented interaction and applies flexibility metrics to generate optimized presentation schedules. But Firefly does not support grouping techniques and a multimedia author might get lost in the specification representation.

Mode [BHL91] (Multimedia Objects in a Distributed Environment) was designed to address the problem of presenting multimedia documents in distributed heterogeneous hardware environments. It supports media segments called objects that have fine granularity, predictable durations, continuous adjustability, discrete adjustability, and flexible metrics. Further, it supports temporal relationships called synchronization points that may be placed between two or more events. Interactive objects allow readers to control a documents presentation dynamically.

Mode includes a graphical authoring tool called the synchronization editor which allows the interactive specification of synchronization relations, temporal and spacial layout. Although the editor provides different views, the synchronization model is not flexible and authoring on synchronization point level might overtax unprofessional authors.

Mode's hybrid formatter provides a compile-time component called the Optimizer and a run-time component called the Synchronizer. The input of the formatter consists of a manually-constructed layouts with an environment specification. The optimizer uses the environment specification to select a presentation quality to be used for this presentation of the document. The synchronizer performs the synchronized presentation of objects by handling media delays via a runtime signalling mechanism.

CMIF [HRB93] addresses the problem of presenting multimedia documents in a heterogeneous hardware environment. It supports media segments, called data blocks, that have coarse granularity, predictable durations and no flexibility. In CMIF button-driven interaction can be defined on various levels. To add temporal constraints to multimedia documents, two separate views are used: A hierarchical view represents graphically the rough hierarchical structure of the document using two temporal relations: sequential and parallel. Since those temporal relations are insufficient to represent all temporal constraints, supplementary temporal relations called synchronization arcs can be added in a channel view. A channel is a kind of time-line which contains a sequence of the media items sharing the same resource, e.g. the same audio channel. Like any other time-line model CMIF is not capable to model uncertainty and it might be confusing for the author to specify the temporal layout in two different views. CMIF's run-time formatter is called player and uses a temporal dependency graph traversal algorithm.

In summary, existing systems focused on enabling authoring of multimedia titles and the synchronization of its media elements. Interaction was subsidiary goal if at all, and most systems have not an integrated temporal interaction model. Further, many system features are implementation platform specific not integrating a general model for a variety of systems.

7. Summary

The increasing range of application for multimedia presentations such as teaching, education or advertising requires enhanced concepts of interactivity and synchronization to guarantee their acceptance. *Tiempo* is a multimedia authoring and presentation system providing easy handling of distributed, interactive multimedia titles. The system contains authoring- and presentation-tools as well as translators to integrate documents in standardized platform-independent notations. The *Tiempo* system is designed modular to be able to test alternative techniques or algorithms and to simplify the adaption to various heterogenous system platforms. Various system mechanisms such as scratching and continuous consistency checking support an author during the interactive composition of a multimedia document. The graphical editor is based on a developed multimedia document language providing features such as templates, hierarchical structuring, abstract, indeterministic temporal synchronization constraints and multiple interaction forms. The run-time system optimizes the presentation of a document by dynamically resolving indeterminism in respect to resource changes, interaction and presentation faults. The goal of this optimization is to keep the outlined synchronization constraints by minimal resource requirements.

The implementation of *Tiempo* is in progress. Currently, we complete the first prototype of the multimedia editor and the *Tiempo* language. The implemented hybrid run-time system is not optimized so far and a lot of work has to be done in the field of the algorithms resolving indeterminism dynamically in respect to interaction or system events.

8. References

[BDH+93] Ingo Barth, Gabriel Dermler, Tobias Helbig, Kurt Rothermel, Frank Sembach, and Thomas Wahl. CINEMA: a Configurable INtEgrated Multimedia Architecture. In *GI/ITG-Arbeitstreffen*, 2 1993.

[BHL91] G. Blakowski, J. Huebel, and U. Langrehr. Tools for Specifying and Executing Synchronised Multimedia Presentations. *2nd. Intl. Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg*, 11 1991.

[BuGo94] John F. Buford and Chetan B. Gopal. Standardizing a Multimedia Interchange Format: A Comparison of OMFI and MHEG. In *IEEE 1st Intl. Conference on Multimedia Computing and Systems, Boston*, pages 463–472, 5 1994.

[BuZe92] M. Cecelia Buchanan and Polle T. Zellweger. Scheduling Multimedia Documents Using Temporal Constraints. *Proc. of the 3rd International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego*, pages 223–235, 11 1992.

[BuZe93] M. Cecilia Buchanan and Polle T. Zellweger. Automatic Temporal Layout Mechanisms. In *ACM 1st Intl. Conference on Multimedia, Anaheim*, pages 341 – 350, 8 1993.

[Drap93] George D. Drapeau. Synchronization in the MAEstro Multimedia Authoring Environment. In *1st Intl. ACM Conference on Multimedia*, pages 331 – 340, 8 1993.

[EFI94] Lynnae Ehley, Borko Furht, and Mohammen Ilyas. Evaluation of Multimedia Synchronization Techniques. In *IEEE 1st Intl. Conference on Multimedia Computing and Systems, Boston*, pages 514–519, 5 1994.

[GKKZ92] K. Gottheil, H. Kaufmann, T. Kern, and R. Zhao. *X und Motif: Einfuehrung in die Programmierung und das X-Windows Systems*. Springer Verlag, 1992.

[HCD+94] I. Herman, G.S Carson, J. Davy, D.A. Duce, P.J.W. ten Hagen, W.T. Hewitt, K. Kansy, B.J. Lurvey, R. Puk, G.J. Reynolds, and H. Stenzel. PREMO: An ISO Standard for a Presentation Environment for Multimedia Objects. In *ACM 2nd Intl. Conference on Multimedia, San Fransisco*, pages 111–118, 10 1994.

[HRB93] Lynda Hardman, Guido v. Rossum, and Dick C.A. Bulterman. Structured Multimedia Authoring. In *ACM 1st Intl. Conference on Multimedia, Anaheim*, pages 283–289, 8 1993.

[HyTi92] HyTime. Information technology - Hypermedia/Time-based Structuring Language (HyTime). *ISO/IEC 10744*, 8 1992.

[Kale93] Kaleida. Kaleida, ScriptX, and the Multimedia Market. Kaleida Labs, Inc., 1055 B Joaquin Road Mountain View, USA CA94043, 1993.

[KeLo91] Somnuk Keretho and Rasiah Loganantharaj. Qualitative and Quantitative Time Interval Constraint Networks. *Proc. of ACM, San Antonio*, 1991.

[MHEG94] MHEG. *DIS 13522- Information Technology - Coding of Multimedia and Hypermedia Information*. ISO/IEC, 8 1994.

[MPEG93] MPEG. Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s. Technical report, ISO IEC JTC 1, IS 11172, 1993.

[PrBo94] Nikos B. Pronios and Theodoros Bozios. Multmedia and Hypermedia Synchronization: A Unified Framework. In Ralf Steinmetz, editor, *Intl. Workshop on Multimedia: Adavanced Teleservices and High-Speed Communication Architectures IWACA'94, Heidelberg FRG*. Springer, 9 1994.

[Pric93] Roger Price. MHEG: An Introduction to futur International Standard for Hypermedia Object Interchange. In *ACM 1st Intl. Conference on Multimedia, Anaheim*, pages 121–128, 8 1993.

[WaRo94] Thomas Wahl and Kurt Rothermel. Representing Time in Multimedia Systems. In *IEEE 1st Intl. Conference on Multimedia Computing and Systems, Boston*, pages 538–543, 5 1994.

[WRW95] Stefan Wirag, Kurt Rothermel, and Thomas Wahl. Modelling Interaction with HyTime. In *GI/ITG Kommunikation in Verteilten Systemen, Chemnitz, Germany*, pages 188-202, 2 1995.

[WWR95] Thomas Wahl, Stefan Wirag, and Kurt Rothermel. Tiempo: Temporal Modeling and Authoring of Interactive Multimedia. In *IEEE 2nd Intl. Conference on Multimedia Computing and Systems, Washington DC*, 5 1995.