TIEMPO: Temporal Modeling and Authoring of Interactive Multimedia

Thomas Wahl, Stefan Wirag, Kurt Rothermel

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

Keywords

interaction, authoring, multimedia, hypermedia, temporal modeling, synchronization, presentation specification, human computer interfaces, *TIEMPO*[1]

Abstract

Future multimedia titles will provide many ways to interact with the presented information. When creating an interactive multimedia title, the author has to specify the temporal layout and the interaction elements. However, this part of the authoring process is not trivial as the temporal layout and the interaction forms are not independent because some interaction forms affect the presentation schedule of the title. This paper identifies various temporal interaction forms and their temporal dependencies, an integrated model for time and interaction is proposed, and we show how this model is applied to ease rapid authoring of multimedia titles.

1. *TIEMPO*: grant of the Deutsche Forschungsgemeinschaft DFG
Temporal **i**nt**e**grated **m**odel to **p**resent multimedia-**o**bjects
Temporales **i**nt**e**griertes **M**odell zur **P**räsentation von Multimedia-**O**bjekten
Modelo **t**emporal **i**nt**e**grado y **m**ultimedial para **p**resentar **o**bjetos
Modèle **t**emporel **i**nt**é**gré et **m**ultimediel pour **p**résenter des **o**bjets

1. Introduction

Multimedia is one of the fastest progressing and innovative areas of computing [Jain94]. Recently, the focus moves from the traditional presentation to interactive multimedia. When users used to be passive while watching a multimedia title, now they will be able to ask for additional information, walk through a virtual multimedia space or interfere with multimedia objects in any thinkable manner.

Before interacting with a multimedia title, the title has to be produced generally with the assistance of a multimedia authoring tool. Although interactive multimedia is very attractive for the end-user, many authoring tools provide only basic features such as selecting a number of multimedia items and arranging them on a time line. If interaction cannot be specified, user interaction during presentation time is restricted to the default settings of the presentation windows if possible at all.

One reason why interaction cannot be specified by many authoring systems is that theoretically interaction might interfere with any part of the multimedia title. Thus by including interaction in multimedia titles, multimedia authoring becomes quite complex. Also, badly designed interactive titles might contain hidden inconsistencies as nobody can foresee all final effects of interaction when creating a title.

In particular, a number of interaction forms affect the temporal layout of the final multimedia presentation. E.g. if the presentation of a graphical element is terminated interactively, we do not know the final termination instant at authoring time. Thus, an interactive multimedia title does not have a unique temporal layout rather it contains an infinite number of potential temporal layouts. Further as we do not know the termination instant of our media items, we cannot align it on a time axis, which is the traditional approach of defining multimedia schedules. So for specifying interaction, traditional deterministic temporal models such as the time line approach are no longer applicable [WaRo94]. Therefore, the supported temporal interaction - this is interaction that affects the temporal layout of a multimedia title - has a significant impact on the design of multimedia authoring systems.

Currently, we are developing the *Tiempo* multimedia authoring system [WWR95], by which interactive multimedia titles can be created. As modeling and specifying interactive multimedia is crucial part of the *Tiempo*-Editor, we studied temporal interaction forms within multimedia. The results of our studies are presented in this paper.

First, we asked which temporal interaction forms exist and what their major characteristics are. Section 2 gives a classification of the basic temporal interaction forms that we found. In Section 3, we examine how interaction might be integrated in temporal models. Section 4 shows how a user interface can be built for specifying interactive multimedia titles based on the integrated temporal model suggested in the previous section. Section 5 gives an example specification of an interactive multimedia title, and shows how it is transformed to a runable document. Section 6 describes other multimedia authoring systems in comparison to the *Tiempo*-Editor.

2. Classification of temporal interaction forms

Interaction can be any way of a user interfering with the multimedia presentation system. Usually, the presence of interaction possibilities are indicated by buttons, slide bars, pop-up menus, command lines or even words or graphical items that can be activated by a mouse click. Interactive elements can be found in separate control windows, on the border of presentation windows or within the presented information.

Temporal interaction are those interaction forms that affect the final presentation schedule during run time. This section enumerates the basic temporal interaction forms and discusses their major impact on the temporal layout of the multimedia presentation.

Looking at various temporal interaction forms, two basic classes can be identified. The first class comprises interaction forms that affect the presentation context, which is characterized by the number and types of media items currently presented and the position within these items. *Starting* and *stopping* a presentation are examples of context affecting interaction. The second class covers interaction forms that affect the presentation speed such as *slower*, *faster*, *pause*, etc..

2.1 Context interaction

Context interaction forms are realized by a *jump* from an original context to a new context. Various jump types can be identified in respect to the number of potential destination contexts. Leaving a context without establishing a new context is called a *stop* or a jump to a 0-context. A *start* establishes a new context. The new context might be a definite, predefined context or a selected context. In this paper, *start* refers only to interaction on a single, definite context. If a document is chosen from a number of potential contexts and started subsequently, we refer to this interaction as *selection*. Choosing the destination document from a finite number of contexts is called *finite selection*. In a second case, the user selects a document by entering the path and the name of the document. By this mechanism an infinite number of documents can be addressed. Therefore, this interaction is called *infinite selection*.

Interaction type	potential destination documents	semantic
stop	0	stop current presentation
start	1	start presentation
finite selection	n	start one of n presentations
infinite selection	∞	start one of ∞ presentations

Table 1: Context interaction types

Table 1 summarizes the context interaction types. The number of potential contexts affect the system resources necessary to support the interaction type.

Context interaction is used to create a new context or to change the current context. In case of creating a new context, only *star*t and *finite* or *infinite selection* might be applied because no presentation can be stopped as no context exists (figure 1). Changing the current context can be done by jumping to an absolute position of the destination document or to a relative position within the current document. In either case, the current presentation position is replaced by a new position. The *stop* interaction is an absolute jump and can only be applied if the a current context exists. The other interaction forms can be used in all cases.

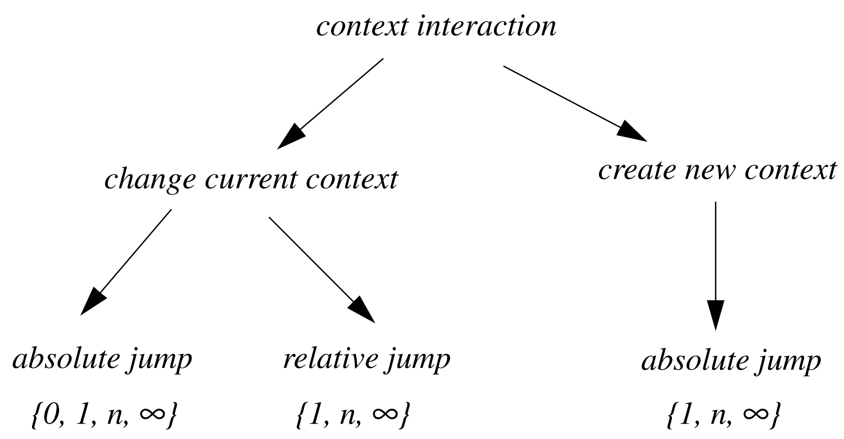

Figure 1: Context interaction

2.2 Speed interaction

Speed interaction modifies the current presentation speed but does not affect the current presentation position. We define the presentation speed of a media item by v = *data/time*. The presentation speed is measured in frames/second, samples/second, etc.. This definition works well with continuous media such as video, audio, etc.. Discrete such as text or graphics do not have a natural presentation speed. However, any of the discrete media such can be considered as a video of identical frames. Anyway, this is the method how discrete media are displayed on the screen. So, displaying a text of 4 minutes equals a video of 7200 identical frames with a presentation speed of 30 frames/second. Thus, doubling the presentation speed, results in displaying the text only for 2 minutes.

A multimedia title might contain several media items. E.g. an animation running at 20 frames/second and a voice at 100 samples/second. So, a complex document does not have one natural speed. We define the presentation speed of a document composed of several media items by the presentation speeds of all media items in the document. Thus, the document presentation speed is a vector of the speeds of the media items. In our example, the presentation speed is a vector of 20 animation frames/second and 100 audio samples/second.

Speed interaction types are *pause*, *continue*, *faster*, *slower*, *reverse*, etc.. The effect of speed interaction is the modification of the presentation speed. This effect can be described by a function mapping the original speed v_{before} to the new speed $v_{new} = f(v_{before})$. Speed functions for specific interaction types are included in table 2. Additionally, table 2 gives the interaction class and its inverse operation for each interaction type. E.g. *pause* is a speed interaction form setting the new speed to zero. To undo the operation, a *continue* has to be executed.

interaction type	presentation speed	class	inverse
stop	not defined	context	start
absolute start, selection	$v_{new} = v_{default}$	context	stop
relative start, selection	$v_{new} = v_{before}$	context	stop
pause	$v_{new} = 0$	speed	continue
continue	$v_{new} = v_{before\_pause}$	speed	pause
faster	$v_{new} = sign(v_{before})* \lvert v_{before} \rvert ++$	speed	slower
slower	$v_{new} = sign(v_{before})* \lvert v_{before} \rvert --$	speed	faster
reverse	$v_{new} = \lvert v_{before} \rvert$	speed	reverse
forward	$v_{new} = -\lvert v_{before} \rvert$	speed	backward $\vee$ none
backward	$v_{new} = -v_{before}$	speed	forward $\vee$ none
set speed	$v_{new} = v_{set}$	speed	set speed

Table 2: Classification of interaction types

Hence, more complex interaction types can be built from the basic temporal interactions in table 2. E.g. on pushing a help-button, a window is displayed with help information. During that time, the main track of the multimedia presentation is paused. So in this example, a compound interaction type is implemented starting the help display and pausing the main presentation.

3. Integrating interaction in temporal models

As temporal interaction affects the presentation schedule, the temporal layout of a multimedia presentation cannot be specified independently from the interaction forms. Therefore, interaction might be included in a temporal model. This section suggests a temporal modelling technique for interaction.

Interaction can be considered as an unpredictable event that might occur during presentation. So at authoring time, we do not know when the user will interact. Consequentially, only temporal models that include uncertainty are applicable for modeling interaction. E.g. deterministic models such as the time line model are not appropriate [MeMo94], [WaRo94]. Indeterministic relational models include uncertainty because only known relations are specified and the author is not forced to specify a relation if it is not predictable.

Temporal models can be based on instants as their elementary unit, e.g. in [BuZe93], [BuZe92], or on intervals [WaRo94]. In this paper, we use the more powerful interval-based model. Since no complex temporal relations will be applied, our considerations can be translated to instant-based models. Hence, any interval can be described by two instants, the start and the end of the interval.

In interval-based models, the presentation of a media item is represented by a temporal interval signifying the start and the end of the presentation. Temporal relations can be specified between media items such as before, while, overlaps etc. [Alle83], [WaRo94]. A summary of the [WaRo94] interval-based temporal model is given in the appendix.

Interaction can embedded in this model because generally it is only supported by the system during some time span. Therefore, interaction might be represented by an interval with a start indicating that interaction is supported from this point on in time and an end signifying that the interaction element is no longer supported or the interaction is activated by the user. So, an interaction interval represents the time during which the corresponding interaction element, e.g. a button, is supported.

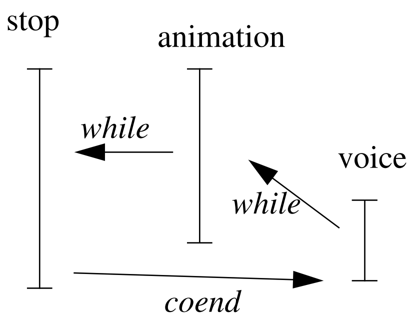

Figure 2: Using interaction intervals for specification

Temporal relations between interaction intervals and media intervals are the same as between media items. Figure 2 shows a scenario in which a voice explanation is given while an animation is presented. The entire presentation can be stopped by pushing a stop-button. The button is present while the animation is active. In this scenario, the animation and the voice are represented by media intervals and the stop-button by an interaction interval.

For the interaction types of table 2, we give a typical temporal representation in figure 3. Variations of the representations might also be used depending on the interpretation of the semantics of an interaction type.

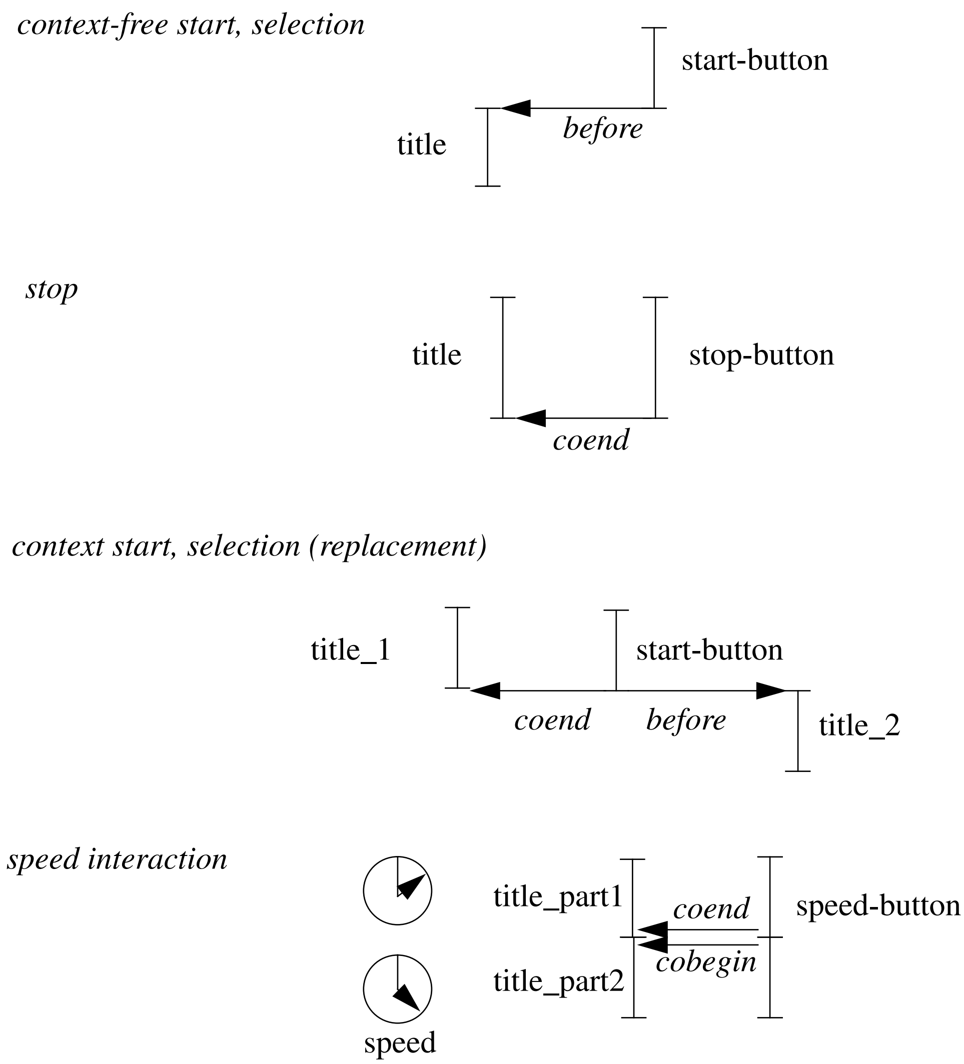

Figure 3: Temporal layout of interaction types

In figure 3, a *context-free start* or *selection* starts the presentation of a multimedia title if it is pushed. The start-button is represented by an interaction interval. If the button is pushed the interaction interval is replaced by the started media interval. The temporal relation between die interaction interval and the media interval is a *before*-relation [WaRo94].

Stop-interaction can be expressed by a *coend* of the stop-button and the media item. A *context start* or *selection* replaces a current multimedia title by a new title. So, the first title is stopped - represented by a *coend* - and the second title is started - represented by a *before*-relation.

Speed interaction can be represented by two phases. In the first phase, a speed changing interaction element is offered while the first part of the title is presented. When the speed element is activated, the second part is presented with a modified speed and generally the speed changing element is continued to be present. In case of reverse-speed operations, the content of the second part is the same as the content of the first part but in reverse order.

Applying a speed changing interaction, the termination instant of the second part might be changed. In a time axis oriented model, this instant and all instant that depend on this instant have to be recalculated all over the entire document. Using an indeterministic relation model only the local speed parameter has to be changed. The modification is propagated automatically on the run without any further changes.

In contrast to deterministic temporal models, this section showed that temporal interaction can be imbedded in a model based on interval relations. By translating intervals into pairs of instants, the same modeling technique can be applied for instant-based models such as in [BuZe93], [BuZe92].

4. Specification by hierarchical grouping

Now, we apply the proposed modeling technique for interaction to specify temporal constraints and temporal interaction in an integrated manner. Due to the presence of interaction, the specification of multimedia titles becomes fairly complex since many new temporal interrelations are introduced specifically if an interaction type affects many media items. In contrary, a comfortable user interface requires simple methods of specifying interaction and temporal relations and it is due to the system to interpret the specification.

Additionally applying the raw version of the proposed model, we cannot conclude from the qualitative structure which interaction type is specified. E.g. various types of speed interaction cannot be distinguished unless we look at the presentation speed of the media items. Therefore, it is easier for the multimedia author to use predefined interaction macros instead of specifying them manually.

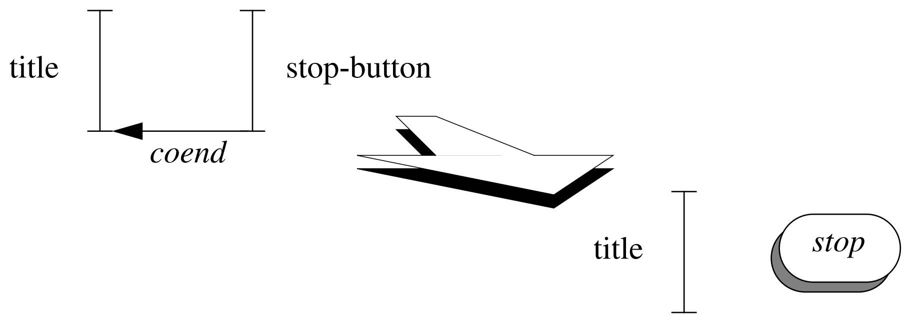

Figure 4: Using interaction macros

Figure 4 shows how a macro specification replaces the explicit temporal specification. The interaction interval is represented by a symbolic interaction element, and the temporal relations between the interaction interval and the media items do not have to be specified explicitly because they are added by the authoring system.

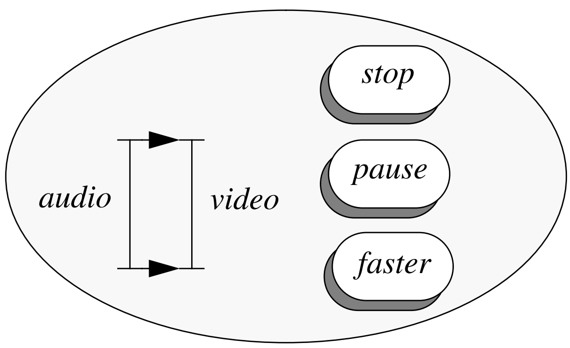

Figure 5: Grouping of media items

Further, an interaction event might affect many media items, e.g. the semantics of a *stop* might be to stop a presentation session. Thus, all media items have to be stopped. A naive way of specifying this scenario would require to enumerate all potential media items that are present when the *stop*-button is pushed. To avoid this enormous specification overhead, interaction might be specified for an entire group of media items. E.g. in figure 5, the buttons affect both, the video and the corresponding audio presentation.

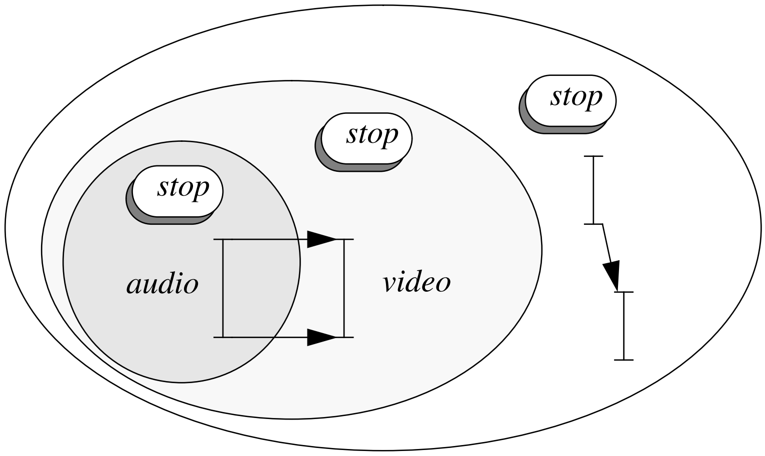

Figure 6: Hierarchical structuring of multimedia titles

In complex multimedia documents, hierarchical grouping of media items is appropriate to specify interaction on various scope levels (figure 6). So, the stop-button is offered with different semantics on different levels, e.g. stop the entire session, stop the video/audio presentation or stop the audio only. Hierarchical grouping seems to be sufficient as we have not observed that media items are shared by several groups. The implementation of a hierarchical grouping mechanism can be done using the same principles as in drawing tools.

5. Scenario

This section provides a simple specification example to discuss how authoring is done in *Tiempo* (figure 7). First, the author selects the media items ‘animation’ and ‘voice’ and defines their temporal interrelation. Then, interaction forms are added such as a possibility to stop this presentation part and a pause/continue macro, which pauses the presentation and offers a button for continuation. The author groups this scene, adds a real time clock animation to the presentation and a stop-button for canceling the entire presentation.

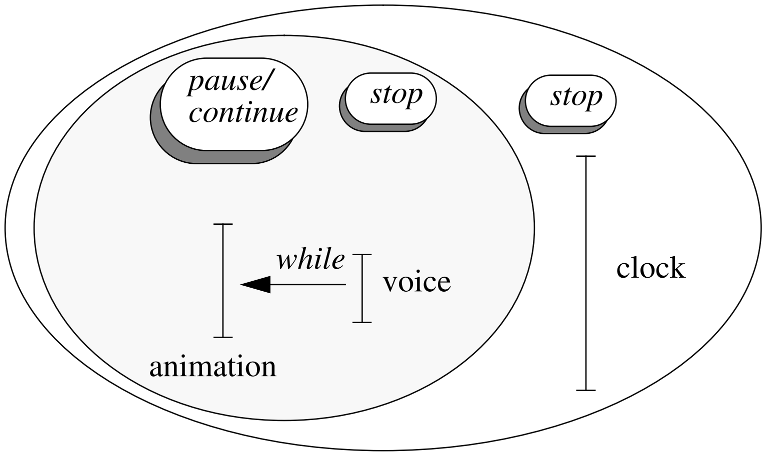

Figure 7: Scenario

Here, we can observe the flexibility of the *TIEMPO* system. A stop-button can be defined for a part of the presentation. So, the animation/voice group can be stopped without affecting the clock. Also by using the graphical *TIEMPO*-Editor, the scenario can be specified rapidly by a few mouse clicks.

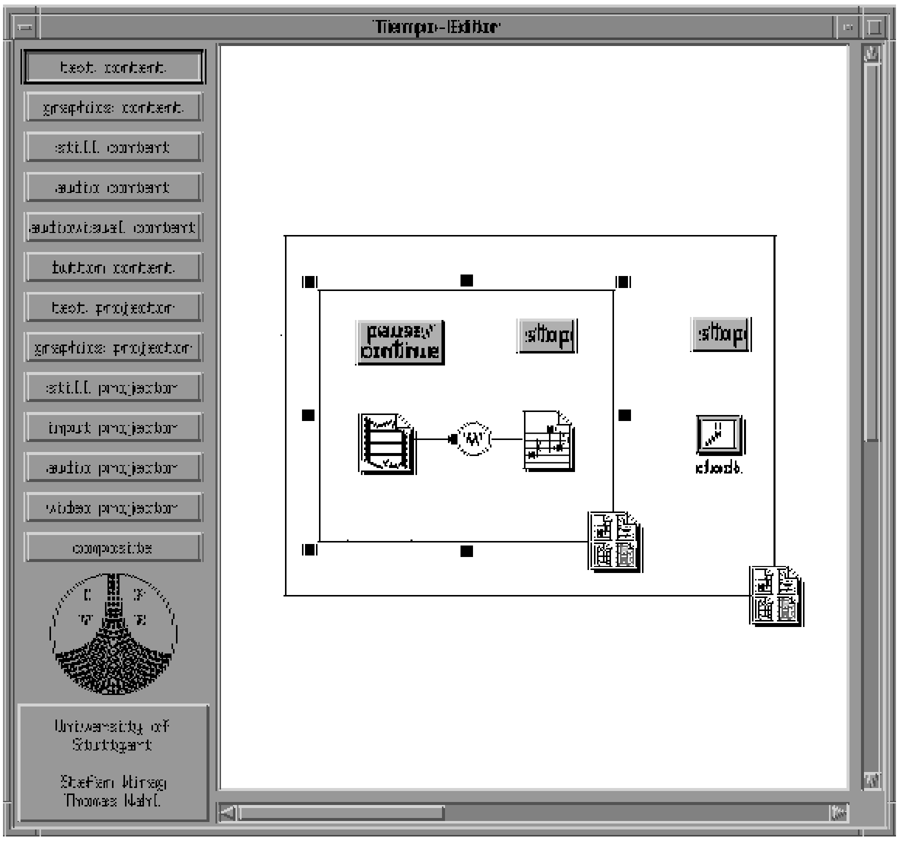

Figure 8: Tiempo - Editor

Figure 8 shows how the scenario is specified in the *Tiempo*-Editor. The media items and interaction elements are represented by icons, and relations by a tagged line connecting the items and elements that are affected by the relation. Groups are visualized by a composite object which can be expanded by a mouse click to look at the interior elements of the group.

Other presentation attributes, such as coding parameters of the media data, spacial extension, volume, etc., can be specified in pop-up menus, which are accessible by clicking the media items. However, these parameters do not have to be specified in *Tiempo* because the systems completes the presentation information by using default templates or attributes that can be derived from the media items, e.g. the coding parameters. Templates are parameters set that can be reused for many presentations giving them a unified layout.

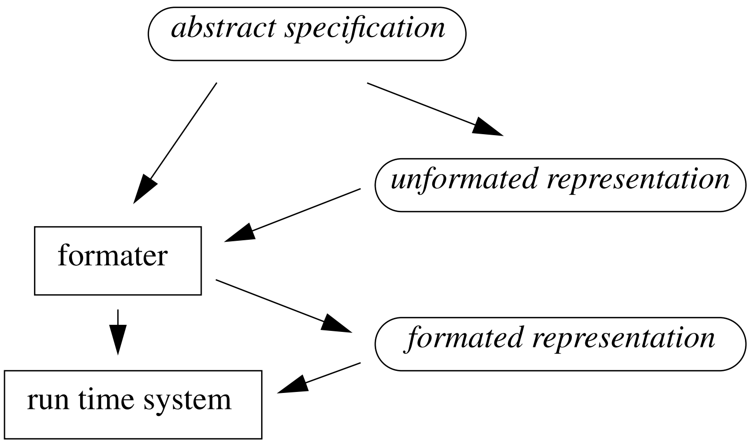

Figure 9: Processing the abstract specification

Subsequently, the abstract graphical specification is either fed to a formater (figure 9) to resolve unspecified layout details or coded in a language for unformated multimedia documents such as the standard language [HyTi92] or one of its extensions [KRRK93], [BRR94]. Coding techniques and extensions are described in [WRW95], [KRRK93].

If the document is formated, it can be run by a multimedia presentation system or stored coded in language for formatted multimedia documents such as MHEG (Multimedia Hypermedia Expert Group) [MHEG94]. The related standardized document and scripting languages ScriptX [Kale93], OMFI [BuGo94], PREMO [HCD+94] are applicable in this context as well.

Non-standardized representation techniques for temporal information are found in [LiGh90], [Hoep91], [AbKl94] et al..

6. Related Work

Previous multimedia authoring systems (MAEstro [Drap93], Firefly [BuZe93], [BuZe92], Mode [BHL91], CMIF [LHBu93], [HBR93], TemporalGlue [HSR94], Quicktime Movie Tool [Appl91]) focused on enabling authoring of multimedia titles and the synchronization of its media elements. Interaction was subsidiary goal if at all. E.g. although MAEstro [Drap93] has features such as distribution and synchronization mechanisms, it does not provide any mechanism to specify interaction. Its interaction is limited to the buttons offered by the display windows.

Firefly [BuZe93], [BuZe92], a conceptually advanced authoring systems, provides a mechanism to associate interaction forms with single media items. However, no grouping or hierarchical structuring mechanism is supported. Thus, multimedia authoring becomes quite complex and an unskilled author might get lost in a huge network of media items, interaction elements and temporal interrelations. Specifically if a title contains a large number of interaction elements, e.g. elements to look up lexicographical media items, a generic macro mechanism is more advantageous.

In the hypermedia-based CMIF[2] authoring system [LHBu93], [HBR93], 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 raw structure of the document using two temporal relations: *sequential*, which is a subtype of the before-relation in [WaRo94] and *parallel* which is subtype of the while-relation in [WaRo94]. Since those temporal relations are insufficient to represent all temporal constraints, supplementary temporal relations can be added in the second view, the channel view. There, a channel contains a sequence of all media items sharing the same resources, e.g. the same window or the same audio channel etc.. Therefore, the channel view is a track and time-line-based visualization of the temporal document layout. However according to [MeMo94], it lacks the capability of modeling uncer-

2. Centrum voor wiskunde en informatica Multimedia Interchange Format

tainty just like any other time line model. Further, it might be confusing for the author to specify the temporal layout in two different views and not getting the whole picture.

7. Summary

Interaction is a big challenge for future multimedia presentation and authoring systems. Therefore, we examined various interaction forms and showed how temporal interaction interferes with the temporal specification of multimedia titles. Applying this knowledge, we have integrated temporal interaction in a temporal model. This will enable multimedia authors to specify temporal constraints and interaction simultaneously respecting their temporal interdependencies. Finally, we applied the model to get a comfortable user interface of multimedia authoring. A maximum amount of system support results in a rapid and easy authoring process. Then, a sketch written by the multimedia author is automatically completed to a multimedia title by the authoring system.

References

[AbKl94] Karl Aberer and Wolfgang Klas. Supporting Temporal Multimedia Operations in Object-Oriented Database Systems. In *IEEE 1st Intl. Conference on Multimedia Computing and Systems, Boston*, pages 352–361, 5 1994.

[Alle83] J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Comm. ACM*, 26(11):832–843, 11 1983.

[Appl91] Computer Inc. Apple. *QuickTime Developer's Guide*. Developer technical Publications, 1991.

[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.

[BRR94] J. F. Buford, L. Rutledge, and J.L. Rutledge. Integrating Object-Oriented Scripting Languages with HyTime. In *IEEE 1st Intl. Conference on Multimedia Computing and Systems, Boston*, pages 456–462. p, 5 1994.

[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.

[HBR93] L. Hardman, D.C.A. Bulterman, and G.V. Rossum. The Amsterdam Hypermedia Model: Extending Hypertext to Support Real Multimedia. *Hypermedia*, 5(1):47–69, 1993.

[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.

[Hoep91] P. Hoepner. Presentation Scheduling of Multimedia Objects and Its Impact on Network and Operating System Support. *2nd. Intl. Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg*, 11 1991.

[HSR94] Rei Hamakwa, Hidekazu Sakagami, and Jun Rekimoto. Mbuild - Multimedia Data Builder with Box and Glue. In *IEEE 1st Intl. Conference on Multimedia Computing and Systems, Boston*, pages 520–525, 5 1994.

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

[Jain94] Ramesh Jain. Multimedia Computing. *IEEE Multimedia*, 1(1), 1994.

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

[KRRK93] John F. Koegel, Lloyed W. Rutledge, John L. Rutledge, and Can Keshin. HyOctane: A HyTime Engine for an MMIS. In *ACM 1st Intl. Conference on Multimedia, Anaheim*, pages 425–434, 8 1993.

[LHBu93] Guido van Rossum Lynda Hardman and Dick C A Bulterman. Structured Multimedia Authoring. In *ACM 1st Intl. Conference on Multimedia, Anaheim*, 8 1993.

[LiGh90] T. D. C. Little and A. Ghafoor. Synchronization and Storage Models for Multimedia Objects. *IEEE Journal on Selected Areas in Communications*, 8(3):413–427, 3 1990.

[MeMo94] S.R.L. Meira and A.E.L. Moura. A Scripting Language for Multimedia Presentations. In *IEEE 1st Intl. Conference on Multimedia Computing and Systems, Boston*, pages 484–489, 5 1994.

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

[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 *accepted for kivs'95: GI/ITG Kommunikation in Verteilten Systemen, FRG Chemnitz 1995*, 2 1995.

[WWR95] Stefan Wirag, Thomas Wahl, and Kurt Rothermel. Tiempo: An Architecture for Multimedia Authoring. In *submitted to Hypermedia - Information Retrieval - Multimedia HIM'95*, 4 1995.

Appendix

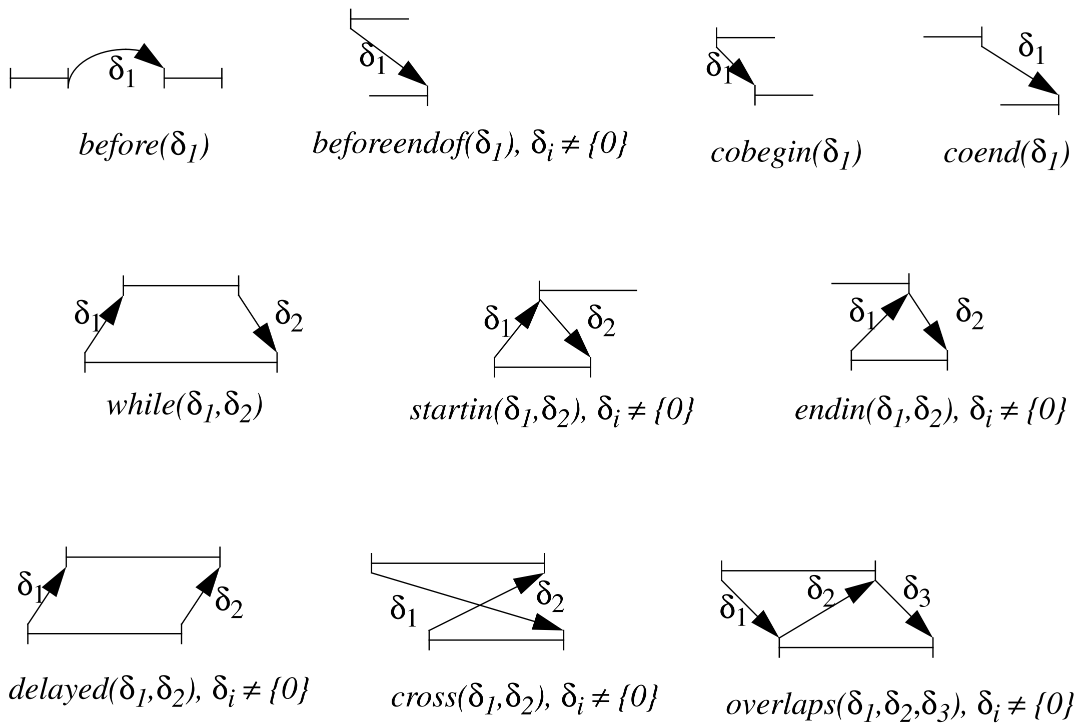

Figure 10: Generic interval relations

Figure 10 summarizes the complete set of the generic interval relations described by [WaRo94]. E.g. if we specify '*media_1 while(0 sec, [0, ∞] sec) media_2*', media item 2 is displayed during the presentation of media item 2. The parameter δ_1 describes the delay between the beginning of media_1 and media_2. In our example, both media start simultaneously because $\delta_1 = 0\ sec$. The second delay parameter δ_2 says how much time after the end of media_2 the media item 1 has to be terminated. In this case, both items might end simultaneously or media_1 ends any time after media_2 has finished because $\delta_2 \in [0, \infty]\ sec$. This relation occurs for example if media_1 has to be terminated interactively.

Look, the interval relations of figure 10 include indeterminism as we do not have to specify precise delays nor any of the exact 13 temporal relations according to [Alle83].