

Adaptive Scheduling of Multimedia Documents

Stefan Wirag

University of Stuttgart
Institute of Parallel and Distributed High-Performance Systems (IPVR)
Breitwiesenstr. 20-22
D-70565 Stuttgart
e-mail: wirag@informatik.uni-stuttgart.de

Abstract

Multimedia presentations are applicable in various domains such as advertising, commercial presentations or education. Multimedia presentations are described by multimedia documents. The presentation of multimedia documents require vast system resources due to the huge amount of data that has to be transferred and processed by the computer system. If multimedia documents can be accessed on-line via different types of networks and be presented on various types of terminals, such as PCs or Set-Top-Units, different amounts of resources may be available at presentation time. Hence, it can happen that there are not enough resources to render a multimedia document according to the specification. For usual multimedia documents resource scarcity implies an arbitrarily reduced quality of the presentation or it can even be impossible to start or continue the presentation. To handle resource scarcity in a better way, multimedia documents can be specified flexible so that they can be adapted to different resource situations. Our temporal model provides abstractions to specify flexible multimedia documents on two levels. It is possible to specify multimedia documents with alternative presentation parts. Further on, the presentation behavior of media objects can vary within specified limits. Hence, the temporal model allows to compose presentations which have a defined behavior when resource restrictions occur. The presented adaptive scheduling algorithm uses the flexibility in specifications to adapt presentations at regular intervals to the current resource situation. Hence, the quality of presentations is reduced or increased in a defined manner.

1. Introduction

Multimedia documents present discrete media objects, such as graphic or text, and continuous media objects, such as video, audio or animation, in an orchestrated manner to the user. In a distributed environment, the structural description of a multimedia document can be stored apart from media object content or even the structural description can be stored in distributed fashion. Hence, the presentation of a document requires not only processing cycles and buffer space but also bandwidth.

For the presentation of a given document various amounts of resources may be available, depending on the type of the presentation terminal, the underlying network as well as the system load at presentation time. For example, consider multimedia documents stored in so-called digital libraries. Ideally, those documents can be accessed via different types of networks and be displayed at various types of terminals, such as PCs, Set-Top-Units or even PDAs.

Even if we consider one type of terminal and one type of network, different load conditions may cause different amounts of resources to be available. If the underlying system provides for resource reservation, the resources needed to present a (single-media) object (e.g. a video clip) can be determined and reserved prior to its presentation. Without reservation the resource situation may change even while the presentation of the object is in progress. However, reservation in general cannot prevent resource shortages to occur during the presentation of multi-object documents. Especially in the case of interactive documents

it is not feasible or even possible to reserve all resources required to render the entire document before presentation starts.

When a resource shortage occurs there are basically three ways to react to it. First, just ignore it and accept the quality of the presentation to be decreased *somehow*. Second, abort the presentation, and third, adapt the presentation in a *user-controlled manner* to the given resource situation. Of course, the third approach requires an appropriate document model as well as an adaptive scheduling mechanism. In this paper, we are going to propose extensions to the temporal model *TIEMPO*¹ [12, 13] to support this type of adaptability and an adaptive scheduling algorithm.

In *Tiempo*, documents are composed of single media objects, such as video, audio or text, and composite media objects, such as pages and scenes. The temporal dependencies between media objects are defined by so called interval operators. Interaction is modeled by reaction relations. The desired adaptability is achieved by *selection groups* which allow to group alternative media objects or presentation parts representing the same information in different form and *Quality of Service ranges* which allow to specify alternative presentation options for media objects. These abstractions can be applied in combination to achieve a high degree of adaptability.

The adaptive scheduling algorithm uses the flexibility in documents to adapt the presentation at regular intervals to the current resource situation. At each adaptation the algorithm tries to update the presentation schedule in such a way that not more resources are needed than available and that the available resources are distributed optimal between simultaneous presented media objects. Thus, resource shortages need not result in an uncontrolled reduced presentation quality of *Tiempo* documents.

The reminder of the paper is structured as follows: In Section 2 we present the basic concepts of the *Tiempo* model. In Section 3, the means that make specifications adaptable are described. Then, we describe our concept of adaptive scheduling. In Section 5 the model used to represent a document specification during the presentation is presented. Section 6 and Section 7 present the adaptive scheduling algorithm. Related work concerning is described in Section 8. Finally, we summarize our results and give an outlook on our future work.

2. Basic Concepts of the TIEMPO model

A multimedia document consist of composite media objects, such as scenes or pages, which are composed of single media objects, such as videos, texts, etc. Single and composite media objects can be classified into *presentation objects* and *interaction objects*. Presentation objects, such as video, audio, text or subpages, present information to the user. Whereas interaction objects, such as buttons, sliders or windows, offer interaction possibilities.

In *Tiempo* a presentation object is modeled by a temporal space, a presentation interval and a projection. The *temporal space (TS)* represents the information of the media object. The *presentation interval* represents the period the media object is presented. It is described by its extent l . The projection describes how information from the TS is presented in the presentation interval. Hence, media objects can be presented with other than recording-time properties. Interaction objects have additional *interaction intervals*. Each such interval represents the period in which a particular user-interaction (e.g. click with mouse, insertion of text) is possible with the media object. An interaction interval is described by its extent and a state that changes when the user triggers the associated interaction.

1. Temporal integrated model to present multimedia-objects

2.1 Temporal Spaces

A TS of a single media object is called *temporal data space (TDS)*. It is defined by the temporally ordered data units of the object, e.g. the TDS of a video object is built by its frames. A TDS is described by a *data interval* with the length n/m . m is the recording speed of the object and n the number of data units. As a discrete media object consists only of one data unit, its TS is by default described by a data interval of length 1.

A TS of a composite media object is called *composite temporal space (CTS)*. It is composed by single or composite media objects. The CTS of the top composite object represents the whole document and its presentation interval the document presentation period. To define the temporal layout of a CTS, presentation intervals of contained media objects are arranged by *interval operators* [12] relative to each other or the CTS.

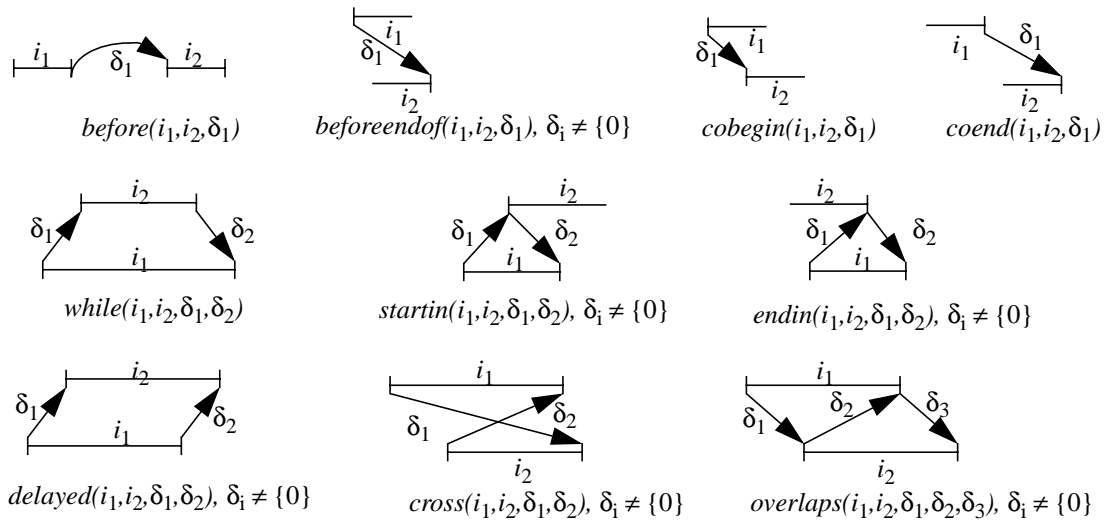


Figure 1: Interval-operators

Figure 1 shows the available interval operators. In the specification example in Figure 2, the interval operator “before” specifies that the animation should be started 1 second after the video has ended. The “cobegin” operator describes that the CTS starts with the video and the “coend” operator describes that the CTS ends with the animation. To define the temporal layout of interaction possibilities, interaction intervals are related to other intervals by interval operators.

2.2 Projection of Temporal Spaces

A projection from a TS to a presentation interval is described by a tuple (v, s) . v defines how many data units of the TS are presented per second within the presentation interval. A presentation speed of 100 implies playback with normal speed. For discrete media objects v is set to 0, which implies that during the presentation always the same data unit is presented.

Parameter s defines which information of the TS is presented in the presentation interval. Most existing models, such as [2, 5], assume that all data units of a media object should be presented. However, it is also possible to present only a part of a media object or to present a media object in a way that there are

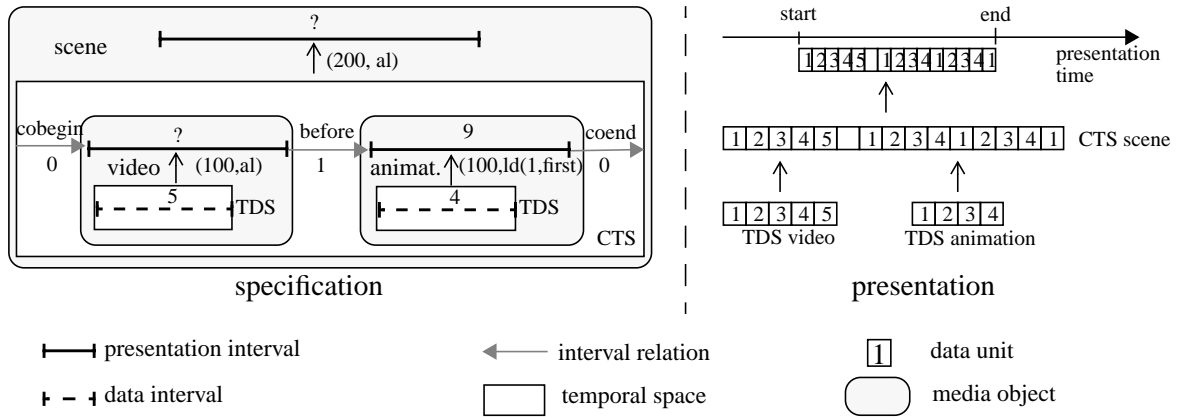


Figure 2: Specification example

less data units than needed to fill the presentation interval. To model such features, our model offers four *alignment policies*:

- *align-length (al)*: The whole TS is presented in the presentation interval. As the extent of the TS is known, it is sufficient to specify parameter v or the extent of the presentation interval l .
- *hold-data (hd)*: The presentation interval can have any extent and any presentation speed can be specified. Gaps at the edges of the presentation interval are filled with the last or first data unit of the TS.
- *loop-data (ld)*: A presentation interval of any length and speed can be specified. Gaps at the edges of the presentation interval are filled by repeating the already presented part of the TS.
- *force-end (fe)*: A presentation interval of any length and speed can be defined, until the TS is large enough to fill the interval.

As the last three policies allow to present only a part of a TS, they have two parameters which define how the TS is aligned in the presentation interval. The first parameter identifies an instant t of the TS and the second parameter defines the position of t within the presentation interval. t can be mapped to the *first* instant, to the *last* instant or to the *center* of the presentation interval. According to this relation and the defined speed, all instants of the TS can be associated with an instant in the presentation interval. With discrete media objects, the policy *hd(1, first)* is used. With stored continuous media objects each policy is possible. The right side of Figure 2 shows how a presentation is created by projection according to the specification on the left side.

2.3 Interaction Relations

In Tiempo, an interaction is described as a so-called *reaction relation* between an interaction interval and affected elements. A reaction relation consists of a trigger-condition and actions. The actions describe behavior modification of projections, presentation or interaction intervals. Actions can start and stop intervals, pause, freeze and speed up the playback [13, 14]. The trigger-condition describes how the state of the interaction interval must change (e.g. a state change of an interaction interval of a button from not-pressed to pressed) before the actions are executed.

3. Modeling of Adaptability in TIEMPO

To be able to adapt documents to different resource situations, flexible specifications are needed. Since flexibility enables the presentation system to create alternative presentations from a document. Considering the structure of a document, flexibility is possible on two levels. Adaptable documents can contain alternative media objects or even scenes which represent the same information in different form. As different information forms have various resource requirements, the presentation system can handle different resource situations. Adaptable documents can contain totally different arrangements of the same media objects. For example, sometimes it makes no difference whether an advertising animation is presented before or after a video-clip. If there are not enough resources to present the media object in one position, it might be possible to present it in another position. In Tiempo, this kind of flexibility is supported by *selection groups*.

Adaptable documents can contain alternative presentation options for the same media objects. For example, it could be specified that a video which is normally presented with 20 frames per second, can alternatively be presented with 10 frames per second. Further, sometimes the starting or terminating instant of a media object can vary within certain limits whether the arrangement to other media objects is not disrupted. To model such flexibility, we introduced *Quality of Service (QoS) ranges*.

Selection groups and QoS ranges can be combined to achieve an optimal adaptability of documents.

3.1 Selection Groups

A selection group consists of an arbitrary number of *presentation alternatives*. It has the semantics that exactly one of the alternatives has to be implemented by the presentation system. Each presentation alternative can consist of several media objects and interval operators. Hence, it can represent any part of a specification. Selection groups can be nested to specify selection possibilities on different levels.

As multimedia presentations should be attractive for users, they often contain media objects which contribute little to the information content. For example, a company logo animation on top of a page. The playout of such information can be omitted when there are not enough resources to present all information. Even some temporal relations can be less relevant. If they are omitted the temporal layout might change so that a resource shortage can be compensated. Scenarios where it is allowed to omit specification parts are modeled by selection groups with an empty presentation alternative.

To limit specification overhead, interval operators relating media objects which are contained in a presentation alternative need not be part of the presentation alternative. The presentation system can detect these interval operators automatically, because if a media object is not presented, interval operators relating the media object to other media objects have no meaning and have to be omitted.

Objects of presentation alternatives of a selection group can be located in different CTSs. In Tiempo, media objects and interval operators are specified within CTSs of composite media objects. Hence, a presentation alternative contains only references to objects.

The presentation system has to know all objects referenced in a selection group to make an adaptation decision. Considering documents where the structural specification is stored in distributed fashion, all referenced objects and the selection group itself have to be specified in a document part that is loaded at once.

A multimedia document may contain several selection groups consisting of multiple presentation alternatives. Hence, the presentation system could have various possibilities to adapt a specification. In Tiempo, priorities can be assigned to presentation alternatives that indicate which alternative should be preferred when more than one can be implemented. The presentation system associates with higher priorities a higher relevance respectively presentation quality. When adapting a presentation, the presentation system will try to implement a combination of presentation alternatives so that the sum of priorities

is as high as possible. 100 is the highest and 0 the lowest priority which can be assigned to a presentation alternative.

So far it is not defined when the presentation system can choose a presentation alternative of a selection group. E.g., sometimes it can be wished that the presentation system continues with the playout of subtitles when the playout of the audio is no longer possible and sometimes not. Especially in interactive documents, scenes can be rendered more than once. It might be desirable that even scenes with presentation alternatives present the same arrangement of media objects in each rendition. In Tiempo, the selection of presentation alternatives is controlled by the following *selection policies*:

- *init*: The presentation system can only once in a document presentation make a selection. If the document part with the selection group is rendered more than once, always the same presentation alternative is implemented.
- *static*: The presentation system can only make a selection at the start of the presentation part which contains the selection group. If the document part is rendered more than once, it can select a different presentation alternative in each rendition.
- *dynamic*: The presentation system can change the implemented presentation alternative during its presentation.

3.2 Quality of Service (QoS) Ranges

A QoS range is a value range where a priority is associated with each value. In the Tiempo model, QoS range arguments can be used to specify the extent of presentation and interaction intervals, the presentation speed and the delays implied by interval operators. If applied, a presentation system can select extent, speed or delay values out of QoS ranges with regard to the resource situation.

A multimedia presentation consist of several media objects arranged by interval operators. Hence, a presentation system can have different options to adapt a specification, e.g. to reduce either the speed of a video or an animation. To indicate which values of QoS ranges are preferred, priorities are assigned to the contained values. A higher priority represents a better presentation quality. When adapting a scene of a presentation, the presentation system should try to select a combination of QoS range values of all objects so that the sum of priorities is as high as possible.

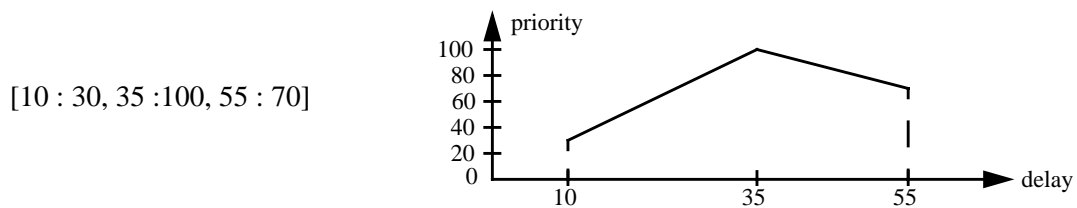


Figure 3: QoS range semantics

The priority structure of a QoS range is defined by arbitrary anchor points. The QoS range in Figure 3 may define the extent of a presentation interval, which can be between 10 sec and 55 sec. In the example, an extent of 10 sec has the priority 30, an extent of 43 sec has the priority 70 and an extent of 35 sec has a priority of 100. With the extents between 10 and 35 linear increasing priorities from 30 to 100 are associated and with the extents between 35 and 43 linear decreasing priorities from 100 to 70 are associated. Here, the presentation system should implement an extent of 35 sec for an optimal presentation

quality. 100 is the highest possible priority. 100 is also the default priority that is associated with values the author has not defined a priority.

If the continuous playout of a media object is a quality criterion, the presentation speed should not change during its presentation. But it can be allowed to choose a particular speed when the media object is started. Further, if scenes are rendered more than once, it might be desirable that some object behavior is equal in each rendition. To define when a value from a QoS range can be chosen, theoretically the same selection policies can be assigned to QoS ranges as to selection groups. To distinguish between static and dynamic selection is only reasonable for speed values, since the user will recognize a dynamic speed change. For interval extents and delays implied by interaction operators a dynamic selection will cause a modified presentation behavior in the future. As the user do not know the future, he cannot distinguish between a static and a dynamic selection policy. Hence, to QoS ranges that define interval extents and delays implied by interaction operators only the selection policies init and dynamic can be assigned.

3.3 Specification Example

Figure 4 shows the temporal specification of a multimedia document according to the Tiempo model. The multimedia document represents a tutorial on a technical process. Selection group *sg1* describes that the presentation can start with a technical animation, a textual description or directly with the video. Selection group *sg2* defines that if the technical animation is presented, simultaneously a speech sequence or subtitles have to be presented. The selection policy of *sg2* is dynamic. Hence, the presentation system can switch between subtitles and speech during the presentation. The selection policy of *sg1* is static. This means, it is not allowed to stop the playout of a presentation alternative and continue with another. During the whole scene an animation of the company logo is presented. As the logo animation has a low relevance, selection group *sg3* contains an empty presentation alternative and the selection policy dynamic. Thus, the logo animation can be omitted or if already started, its playout can be terminated. The assigned priorities indicate that in *sg1* the playout of the technical animation either with the speech or the subtitles is preferred. In *sg2* the playout of the speech is preferred. The low priority of the presentation alternative with the logo animation in *sg3* cause that the logo animation is terminated first when there is a resource shortage which makes it impossible to present all selected objects.

The speed of the video and the logo animation are specified flexible. The speed of the video and the animation can be between 90% and 100% of its original speed. The priorities of the video speed values are higher than the priorities of the logo animation speed values. Thus, the presentation system will reduce first the speed of the logo animation when there is a resource shortage. The delay between the end of the text or technical animation and start of the video is also specified flexible. It can be between 1 and 3 seconds.

4. Adaptive Scheduling of TIEMPO Documents

The aim of the adaptive scheduling of Tiempo-documents is to implement a presentation so that the presentation quality is as good as possible in the current resource situation. Hence, it has to be determined which alternatives of selection groups and QoS ranges can be implemented with the available resources and how the available resources can be distributed between media objects so that the sum of priorities of the implemented alternatives is as high as possible. The following adaptive scheduling algorithm is designed for environments with a best-effort assignment of resources.

To render a multimedia document, a presentation schedule has to be generated according to the document specification. Such a schedule describes when to start and stop the playout of media objects and how fast continuous media objects are played out. According to these information the stream-synchro-

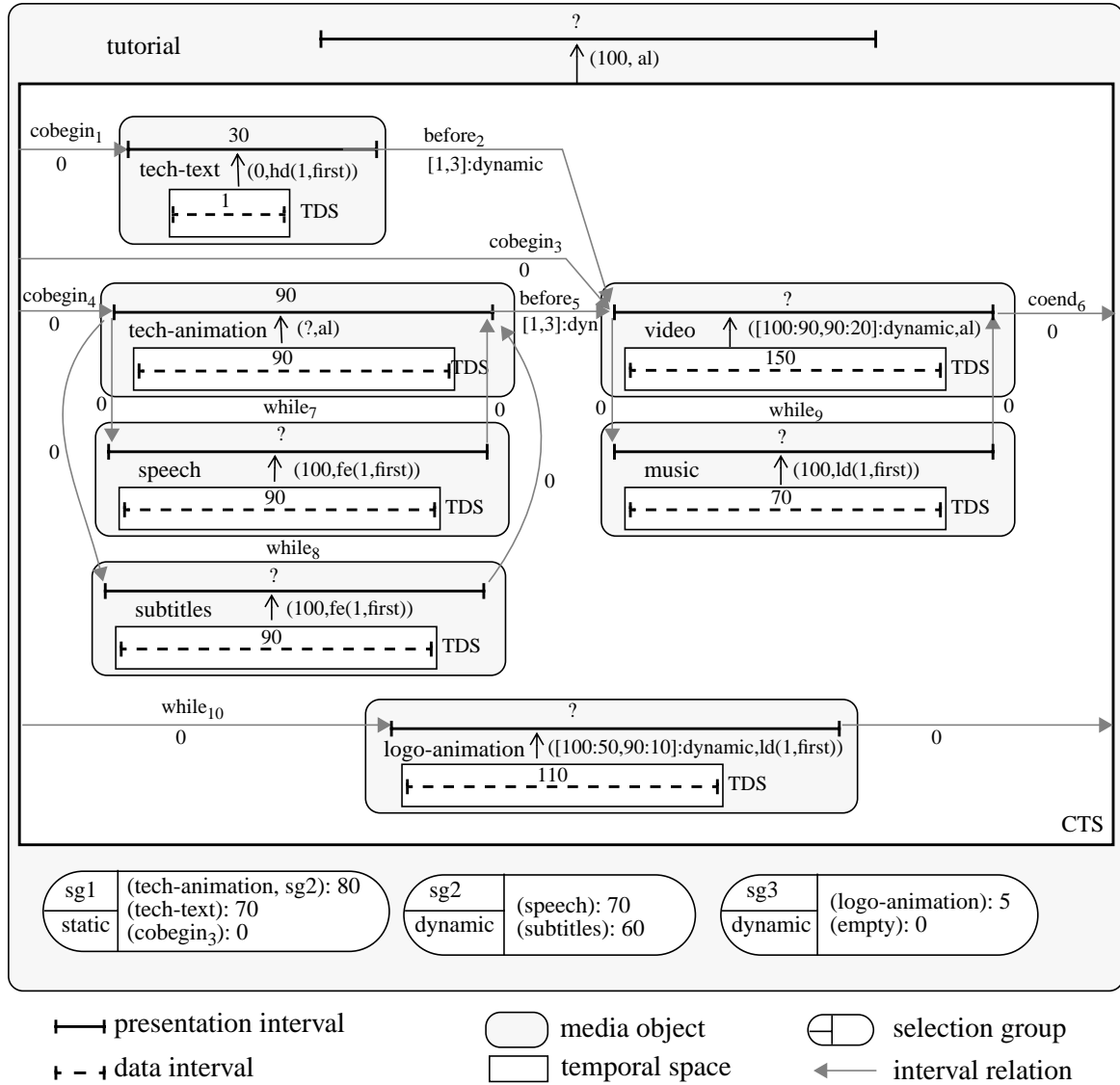


Figure 4: Adaptable tutorial specification

nization mechanism which provides for a synchronized playout of continuous media objects, the transfer of distributed media objects and the playout of media objects is controlled. If the document is interactive the presentation schedule is modified appropriately when interactions occur.

If such a schedule should be generated with regard to the resource situation, we have to consider the following aspects. The resource situation can change during the presentation. Hence, schedules have to be adapted continuously. To be able to distribute the available resources optimal between media objects at an adaptation, predictions have to be made about the amount of resources that will be available in the future. But it is quite hard to make good predictions for long periods. Hence, we suggest the following approach to adapt presentations.

A schedule is adapted at regular intervals during the presentation. (The period between two adaptations is called adaptation interval.) At an adaptation instant, the resource situation is predicted for the next adaptation interval on the basis of the available resources in the recent past. For the period behind the adaptation interval, no assumptions are made about available resources. According to the predicted resource situation, the scheduler creates a schedule for the remainder of the presentation so that there

will not be needed more resources than predicted in the adaptation interval and that the presentation quality is as good as possible. If an adaptation period is over, the scheduler repeats the procedure for the next adaptation interval and so on.

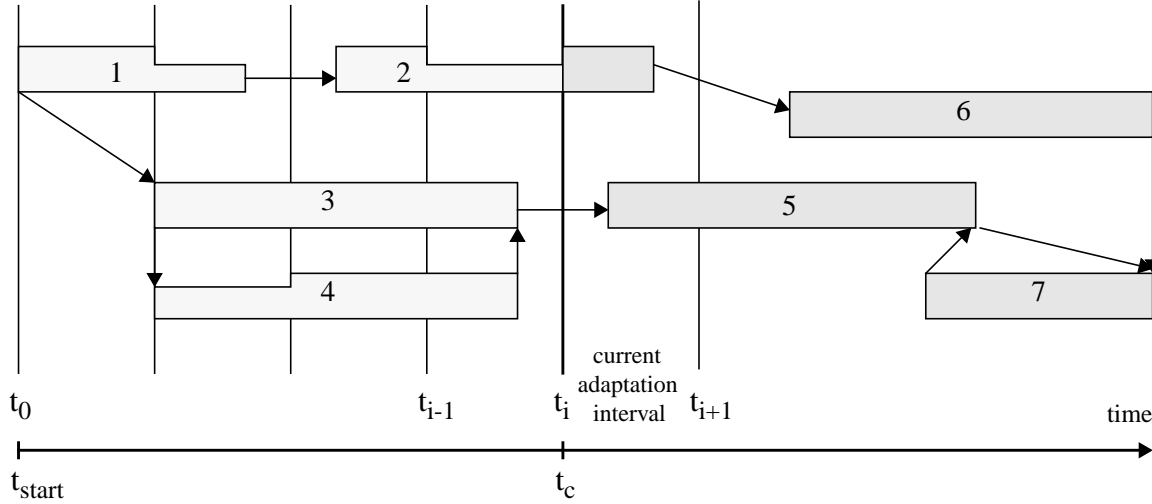


Figure 5: Adaptation scenario

Figure 5 shows an example for the temporal layout of a presentation scheduled according to this approach. The bars represent media objects. The position of the bars shows when media objects are presented and the height of a bar shows how fast a media object is presented. The vertical lines mark adaptation instants. In the example the document has been rendered till instant t_c . Hence, the presentation of media object 1, 3 and 4 has been completed, the presentation of media object 2 is running and media object 5, 6 and 7 will be presented in the future.

Let us assume that the schedule of the scenario should be adapted at t_c and that the specification is flexible. According to the described approach the scheduler will now try to find a presentation speed for the reminder of media object 2, a stop instant for media object 2, start-instants, stop-instants and speeds for media objects 5, 6 and 7 so that in the interval t_i to t_{i+1} not more resources are required than available and that the sum of priorities of the chosen speed, delay and length values is as high as possible.

Adaptation intervals should not be too long, since the reaction to resource changes might become too slow. On the other hand, adaptation intervals should not be too short, to avoid hectic reactions on temporary resource changes which could for example be equalized by the stream-synchronization mechanism. We feel that adaptation periods between 5 and 10 seconds are reasonable.

4.1 Resource Information

To be able to predict the resource situation in the next adaptation interval and to distribute resources between media objects, the following information are needed:

- **Information about resources requirements of media objects**

These information describe which and how many resources a media object requires to be presented in the specified way. On the base of such information it is possible to recognize if there are enough resources. Further, such information are needed to distribute resources between media objects. These information can partly be computed from the specification, partly it have to be determined

experimentally or even assumptions have to be made. For example, the needed bandwidth to transfer a JPEG- or CellB-encoded video can be computed from the frame size and the compression factor. For MPEG-encoded videos this is not possible, because of the varying encoding of frames. Hence, an average data rate has to be assumed. The CPU-load produced when a video is played out with a certain frame size and a certain rate can be determined experimentally.

- **Information about the sharing of resources**

These information tell the scheduler how available resources can be distributed between media objects. CPU cycles and memory are shared between all processes on a system. Hence, it is possible to redistribute CPU cycles and memory between simultaneously presented media objects by reducing the speed of one media object and increasing the speed of another media object. In contrast to this, bandwidth can generally not be distributed in this way. Because it depends on the path that a media stream takes from the source, whether the reduction of the bandwidth requirement of one media object allows to increase the bandwidth requirement of another media object. If the path goes through common gateways and the bottleneck is one of the gateways, a redistribution might have the desired effect. If the paths are completely different except of the end point, a redistribution will in general have not the desired effect.

- **Information about available resources**

Information about the resources which are currently available or that have been available in the recent past can be delivered from resource monitoring components, a management system or components which perform the playout of media objects. On the base of these information the resource situation in the next adaptation interval can be predicted. If information come from playout components, it is not possible to detect how many resources are actually available, since only information about the really used resources and occurring resource shortages can be delivered. To obtain information about the amount of resources actually available, a management system or a resource monitoring component is needed. If such components are not available, assumptions about the available resources have to be made. E.g. we can assume that there are 10% more resources available than currently used, if playout components do not indicate a resource shortage.

4.2 Computation of Adaptations

Based on the resource information, an adaptation can be computed as follows at an adaptation instant: The scheduler determines the best combination of presentation alternatives of selection groups in the remainder of the presentation. This is the combination where the sum of presentation alternative priorities is highest. Then it tries to find for the media objects in the combination and the media objects not contained in selection groups a presentation schedule that does not need more resources than available in the adaptation period and for which the sum of chosen values from QoS Ranges is highest. If this is not possible without violating the specification, the scheduler repeats the same procedure for the next best combination of presentation alternatives and so on. If all combinations require more resources than available, the combination with the lowest violation can be implemented or the presentation can be terminated.

In the following sections a detailed description of this algorithm is given.

5. Scheduling Graphs

The model that is used to represent a document specification during the presentation is a so called *scheduling graph*. A scheduling graph is a directed acyclic graph which is derived from the document specification before the scheduling starts. If a specification contains selection groups, a separate scheduling graph is generated for each possible combination of presentation alternatives.

5.1 Generation of Scheduling Graphs

In a scheduling graph each interval x of a media object i is modeled by two nodes. The node $b_{x,i}$ represents the start-event of the interval and the node $e_{x,i}$ represents its end-event. These nodes are related by an edge which represents the delay between the nodes respectively events. The TS of a media object i is modeled by a node $b_{t,i}$ representing the start-event of the TS and a node $e_{t,i}$ representing its end-event. Figure 6 shows which edges are necessary between the presentation interval nodes and the TS nodes to implement the specified alignment policy of the media object. Delays implied by interval operators are modeled as edges between presentation interval nodes, interaction interval nodes and TS nodes according to the semantics of an interval operator.

The edges in a scheduling graph are labeled with the possible delay values between the nodes. In a Tiempo specification media objects are nested within each other. In each TS of a media- or multimedia object the time may pass with different speed. The specified speed value of the projection describes how fast the time should flow in the TS of a media object relative to the TS in which the media object is contained. As the nodes of the scheduling graph describe instants of the real time, specified delay values have to be divided by the effective speed in the TS they are associated with to gain the correct delay value for the scheduling graph. The effective speed $v_{e,i}$ within a TS of media object i is defined recursively by $v_{e,i} = v_i v_{e,j}$ where v_i is the speed specified for media object i and $v_{e,j}$ is the effective speed of multimedia object j that contains media object i . Figure 6 shows how the delay values of the edges are computed within a media object according to the specified alignment policy. In Figure 6, parameter l represents the specified length of the presentation interval, parameter v represents the specified speed of the media object and t represents the instant used to arrange the TS of the media object in the presentation interval.

When speed values, interval lengths and delays implied by interval operators are specified as QoS ranges, edges will be labeled with a value range which contains all possible delays.

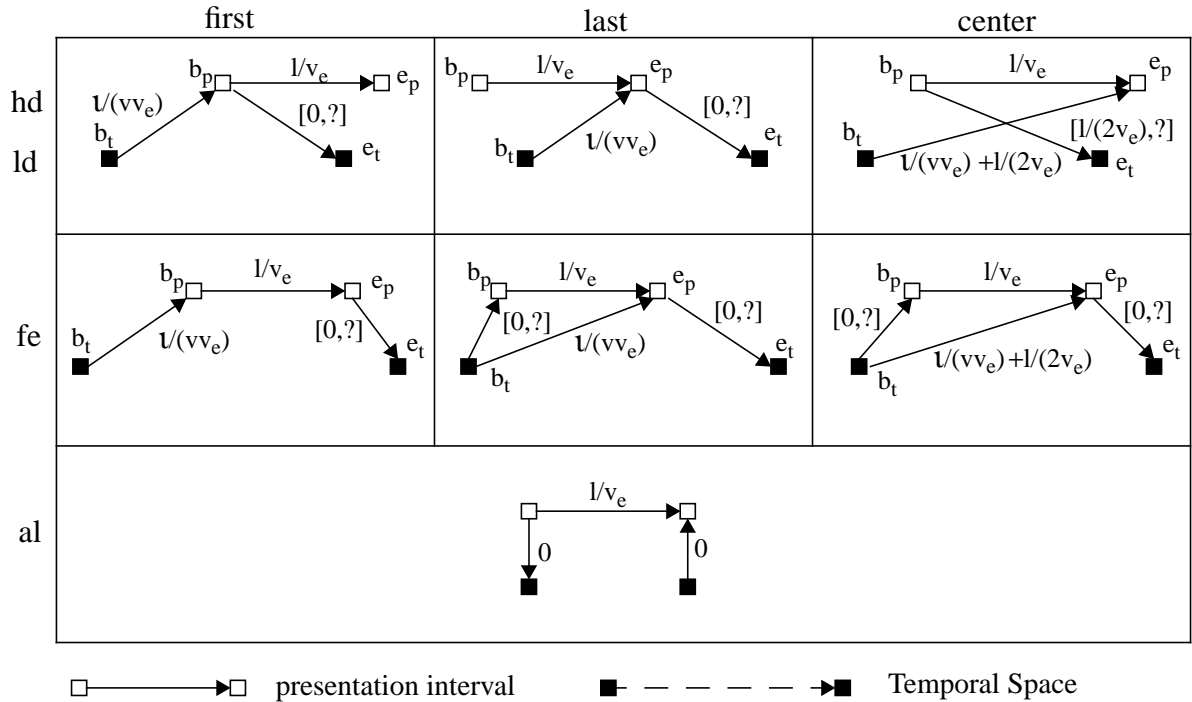


Figure 6: Representation of media objects in scheduling graphs

5.2 Computation of Event Ranges

When the scheduling graphs have been generated, we compute for each scheduling graph the instants at which the events represented by the nodes can occur. This information is necessary to detect which media objects have to be considered at an adaptation instant. The set of instants at which an event can occur is called *event range*.

The event ranges of a scheduling graph are computed applying the following algorithm:

1. **Enumeration:** The nodes of the graph are enumerated so that a node gets a higher number than its predecessors. The event range of the node with the lowest number is set to $[0]$ because this event represents the start of the whole presentation. The event range of all other nodes are initialized with $[0, ?]$ which indicates that the event can occur between the start of the presentation and the end of the presentation which is unknown.
2. **Forward Computation:** The event ranges $N(j)$ of all nodes are corrected by applying the formula

$$N(j) = \bigcap_{i \in \text{predecessor}(j)} (N(i) + D(i, j)) \cap N(j)$$

in order of the ascending enumeration of the nodes. $D(i, j)$ is the delay implied by the edge from node i to node j . The term $N(i) + D(i, j)$ represents the set of instants that we get if we add to each instant in $N(i)$ each instant in $D(i, j)$.

3. **Backward Computation:** The event ranges $N(j)$ of all nodes are corrected by applying the formula

$$N(j) = \bigcap_{i \in \text{successor}(j)} (N(i) - D(i, j)) \cap N(j)$$

in order of the descending enumeration of the nodes.

When this algorithm has terminated, all event ranges contain the possible instants. Hence, we know now exactly when an event can occur.

6. Adaptation Computation

To find the optimal adaptation at an adaptation instant t_k , we apply the algorithm described in the remainder of this section to the scheduling graph representing the best combination of presentation alternatives. If the algorithm is successful, we have found the optimal schedule in the current resource situation. If it fails, the procedure is repeated for the next best scheduling graph and so on.

For the following description we assume that we are in the middle of a presentation. This means, there have already been a couple of adaptations.

6.1 Detection of Possible Media Object Overlapping

In an adaptation interval different overlappings of media objects might be possible. As a different overlapping of media objects implies various resource requirements, we have to check for each different overlapping scenario if there are enough resources. If an overlapping scenario does not require more resources than available, we compute the combination of start- and stop-instants of media objects and speeds of continuous media objects which has the highest sum of priorities. The overlapping scenario with the highest sum of priorities is then the optimal schedule of the presentation.

To detect the possible overlapping scenarios, we determine on basis of the event ranges the presentation interval nodes respectively events which could lie in the adaptation interval. Then, a matrix called *Event Relation Matrix* (ERM) is generated which describes all possible temporal relations between these nodes and between these nodes and the boundaries t_k and t_{k+1} of the adaptation interval. A ' $<$ ' in a cell of an ERM means that an event has to occur before an other. A ' $>$ ' in a cell means that the an event has to occur after an other.

In a ERM we insert a ' $<$ ' in the cell which describes the relation between two nodes a and b , if the scheduling graph has an edge from node a to b . If the scheduling graph has an edge from b to a , a ' $>$ ' is inserted in the cell. For nodes which are not related directly by edges we take a look at the event ranges. If the event ranges of two nodes have common instants, we insert a ' $<$ ' and ' $>$ ' in the cell of the matrix. If there are no common instants, we insert a ' $<$ ' or a ' $>$ ' dependent on which event range contains smaller values. If a node can lie at the current adaptation instant t_k we insert a ' $<$ ' in the appropriate cell. If it can lie behind t_k , we insert a ' $>$ ' in the appropriate cell. If a node can lie behind the next adaptation instant t_{k+1} , we insert a ' $>$ ' in the appropriate cell. If it can lie before t_{k+1} , we insert a ' $<$ ' in the appropriate cell.

For example, let us assume that we have an adaptation scenario as shown in Figure 7. In the example the specification of the document is represented as scheduling graph. In this example the document was presented till instant t_2 which is the current adaptation instant. Hence, the instants of the nodes which lie before t_2 are fixed. Now we want to compute an adaptation for the adaptation interval $t_2 - t_3$. Let us assume that the node $e_{p,3}$ and the node $b_{p,4}$ can lie before or after t_3 . Then the ERM would look like Table 1.

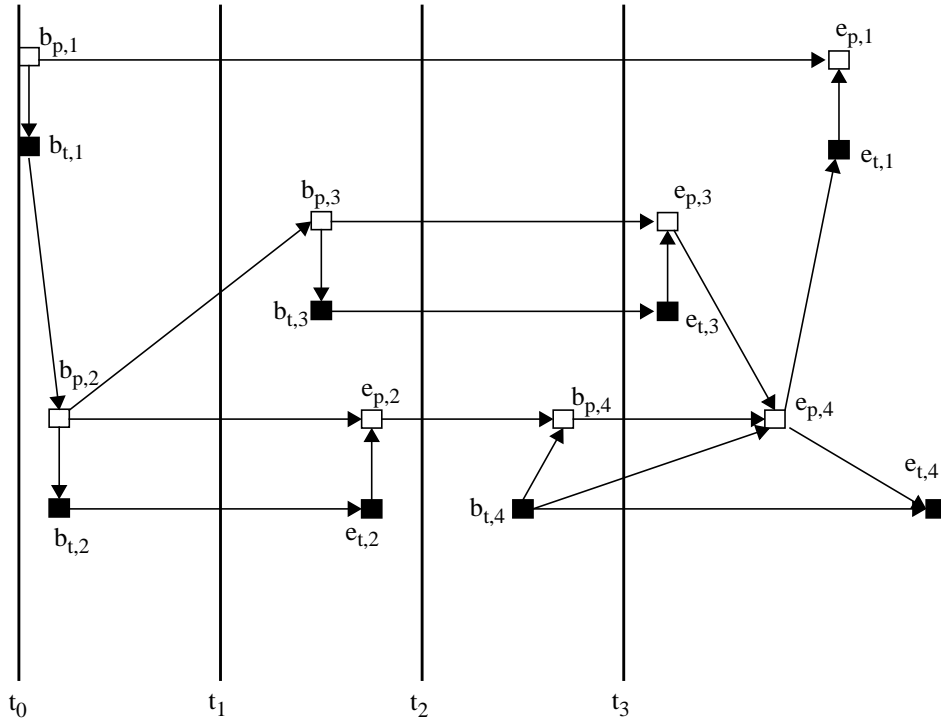


Figure 7: Adaptation scenario

One particular overlapping scenario can be derived from an ERM by choosing from each cell exactly one relation. By choosing in each stage another possible combination of relations, all overlapping scenarios can be determined. When all overlapping scenarios have been determined, we check with the help of the

	t_3	t_2	$b_{p,4}$
$e_{p,3}$	$<, >$	$>$	$<, >$
$b_{p,4}$	$<, >$	$>$	
t_2	$>$		

Table 1: Event Relation Matrix

given resource information which of them require not more resources in the adaptation interval than available.

Therefore, we have to distinguish between resources that have to be exclusively assigned to media objects and resources that are shared between media objects. An exclusively assigned resource is for instance an audio device which can only be used by one media object at each instant. For such resources it is checked if media objects which need the resource are overlapping in the adaptation interval. If this is the case, the overlapping scenario causes a resource violation. For resources which are shared, we compute for each media object presented in the adaptation interval the minimum resource requirements. The resource requirements of continuous media objects depend on their playout speed. Hence, the minimum resource requirements are given for the lowest playout speed. The resource requirements of discrete media objects are constant. Then, we check if there is an overlapping of media objects in the presentation interval where the sum of the minimum resource requirements is greater than the available amount of resources. If this is the case, the overlapping scenario causes a resource violation.

For each overlapping scenario that does not cause resource violations, a particular scheduling graph called overlapping graph, which represents exactly the overlapping scenario, is derived from the scheduling graph. An overlapping graph contains compared with the scheduling graph event ranges so that the nodes stay in or behind the adaptation interval. Further, it contains additional edges which guarantee that the overlapping of media objects do not change.

To generate an overlapping graph, we make a copy of the scheduling graph which is modified as follows: If the overlapping scenario demands that a node lies in the adaptation interval, we delete the values in the event range of the node which describe instants behind t_{k+1} . If the overlapping scenario demands that a node lies behind the adaptation interval, we delete the values in the event range of the node which describes instants before t_{k+1} . To determine which edges have to be inserted, we take a look at the relations and at the reduced event ranges of the overlapping scenario. If the overlapping scenario demands that a node a lies before b and the reduced event ranges of the nodes have common instants and if there is no edge between a and b , we introduce an edge between a and b labeled with delay value range $[0, ?]$.

Let us assume that we have chosen the overlapping with $e_{p,3} > t_3$, $b_{p,4} < t_3$ and $b_{p,4} > e_{p,3}$ in the example. Then it would be sufficient to delete all instants smaller than t_3 from the event range of node $b_{p,3}$ and to delete the instants greater than t_3 from the event range of node $e_{p,4}$ to generate an overlapping graph.

For each of the generated overlapping graphs, we compute now the optimal speeds, start- and stop-instants with the help of a Mathematical Programme.

6.2 Generation of a Mathematical Programme

A Mathematical Programme is an optimization problem subject to constraints in \mathfrak{R}^n of the form

$$\begin{aligned}
 &\text{Maximize } f(x) \\
 &\text{subject to} \\
 &g_i(x) = 0; \quad i = 1, \dots, m \\
 &x \in S \subset \mathfrak{R}^n
 \end{aligned}$$

The vector $x \in \mathcal{R}^n$ has components x_1, \dots, x_n which are the unknowns or variables of the problem. The function f is called the *objective function* and the set of conditions $g_i(x) = 0$ ($i = 1, \dots, m$) and $x \in S$ is the set of the *constraints* of the problem. The optimal solution of such a problem is a vector x^* that fulfills the constraints and maximizes the objective function. A Mathematical Programme of this form is constructed on the base of an overlapping graph.

The adaptation scenario in Figure 8 is used to illustrate the generation of such a Mathematical Programme. In this example the document was presented till instant t_2 which is the current adaptation instant and we want to compute an adaptation for the adaptation interval $t_2 - t_3$. The position of the nodes shows if they have to lie in or behind the adaptation interval. The position of start- and end-instants before t_2 shows when media objects were started and stopped in the past. The grey rectangles show how arcs are affected by the different presentation speeds of the media objects.

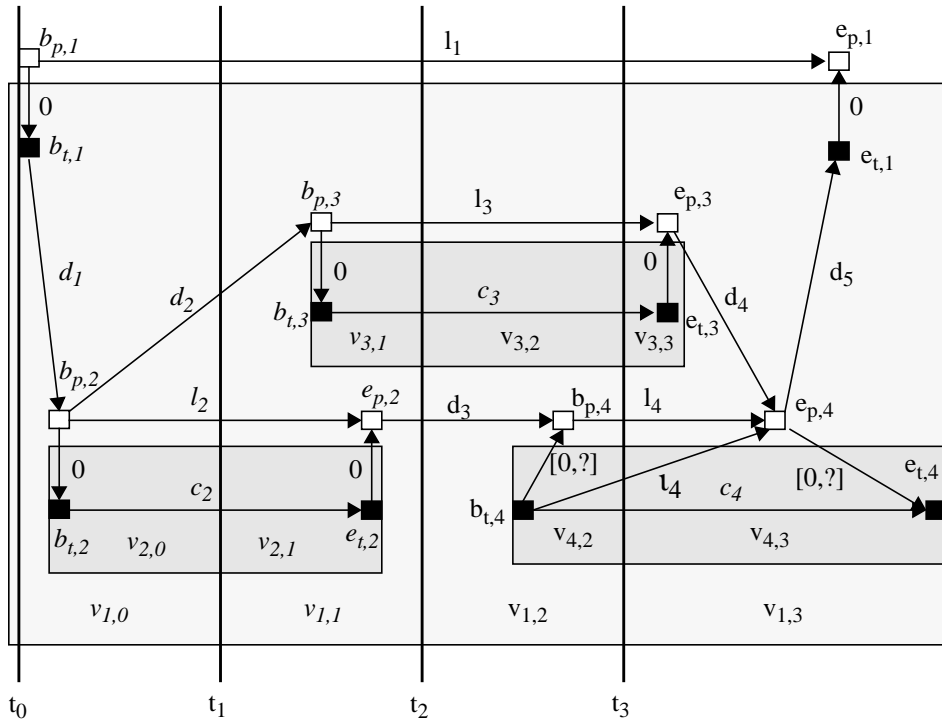


Figure 8: Adaptation scenario

Constraints representing the document specification

Let us assume that all temporal parameter have been defined by QoS Ranges as follows: a delay implied by an interval operator i as $[\pi_{1,d,i} : \rho_{1,d,i}, \pi_{2,d,i} : \rho_{2,d,i}]$, a speed of a media object i as $[\pi_{1,v,i} : \rho_{1,v,i}, \pi_{2,v,i} : \rho_{2,v,i}]$ and the length of a presentation interval of a media object i as $[\pi_{1,l,i} : \rho_{1,l,i}, \pi_{2,l,i} : \rho_{2,l,i}]$. At the end of the section we will describe modifications which are necessary with QoS Ranges of a different form. In the following description parameters which are constants are set in italic style.

To describe an adaptation scenario by constraints, we introduce appropriate variables $b_{x,i}$ and $e_{x,i}$ which represent start- and end-instants of intervals and TSs behind t_k . Further, we introduce variables representing the speed of media objects behind t_k . If the selection policy ‘dynamic’ was specified for the QoS Range defining the speed of a media object, the speed can be different in each adaptation period. If a part

of such a media object i is going to be presented in the adaptation interval, we introduce a variable $v_{i,k}$ which represents the relative speed of the object in the adaptation interval. If a part of such a media object is going to be presented in the region behind the adaptation interval, we additionally introduce a variable $v_{i,k+1}$ which represents the relative speed of the object behind the adaptation interval. For media objects applying the selection policy 'static' or 'init' for speed values we introduce only one variable $v_{i,k}$ if the media object is going to be started behind t_k . Because if it is already running, the playout speed is determined and we are not allowed to modify it. Further, we introduce variables l_i representing lengths of presentation and interaction intervals for intervals where the end-instant lies behind t_k . Variables d_i which represent delays implied by interval operators are introduced when the end-instant lies behind t_k . Figure 8 shows which variables have to be introduced in the scenario.

The variables $v_{i,k}$, l_i and d_i are bounded by the QoS Ranges of the specification. Hence, we introduce for these variables constraints of the following form:

$$\pi_{1,v,i} \leq v_{i,k} \leq \pi_{2,v,i}; \quad \pi_{1,l,i} \leq l_i \leq \pi_{2,l,i}; \quad \pi_{1,d,i} \leq d_i \leq \pi_{2,d,i}$$

As the variables $b_{x,i}$ and $e_{x,i}$ are bounded according to the event ranges in the scheduling graph, we introduce for these variables also appropriate constraints.

With the help of these variables an adaptation scenario can be described by constraints which are constructed according to the edges in a scheduling graph. The following cases have to be distinguished:

- If the start-node b_j and the end-node e_i of an edge h lie both behind t_{k+1} , the following constraint is generated:

$$(e_i - b_j) v_{e,h,k+1} = l_i$$

This equation expresses that the delay between the instants e_i and b_j under consideration of the effective speed $v_{e,h,k+1}$ in the TS the edge h is located in has to be equal to l_i . $v_{e,h,k+1}$ is a place holder for the term $v_{a,k+1} v_{b,k+1} \dots$, which is the product of the presentation speeds of all objects which contain the edge directly or indirectly. l_i is the variable that represents a value of the QoS Range specified by the author for the delay between e_i and b_j .

In the example the edge between $e_{p,4}$ and $e_{t,1}$ has to be described in this way:

$$(e_{t,1} - e_{p,4}) v_{1,3} = d_5$$

If both nodes related by an edge lie in the adaptation interval, a constraint of the same form has to be generated, but the variable $v_{e,h,k}$ has to be used, instead of $v_{e,h,k+1}$.

- If the start-node b_j of an edge h lies in the adaptation period and the end-node e_i lies behind the adaptation period, constraints of the following form are generated:

$$(t_{k+1} - b_j) v_{e,h,k} + (e_i - t_{k+1}) v_{e,h,k+1} = l_i$$

In the example the delay between $b_{p,4}$ and $e_{p,4}$ has to be described in such a way:

$$(t_3 - b_{p,4}) v_{1,2} + (e_{p,4} - t_3) v_{1,3} = l_4$$

- If the start-node b_j of an edge lies before t_k in the adaptation period $t_{k-g} - t_{k-g-1}$ and the end-instant e_i lies in the adaptation period, the following constraint is generated:

$$(t_{k-g} - b_j) v_{e,h,k-g-1} + (t_{k-g+1} - t_{k-g}) v_{e,h,k-g} + \dots + (t_k - t_{k-1}) v_{e,h,k-1} + (e_i - t_k) v_{e,h,k} = l_i$$

In this constraint the start-node is not a variable, since if the start-node lies before t_k it is already determined.

- If the start-node b_j of an edge lies before t_k in an adaptation period $t_{k-g} - t_{k-g-1}$ and the end-instant e_i lies behind the adaptation period, the following constraint is generated:

$$(t_{k-g} - b_j) v_{e,h,k-g-1} + (t_{k-g+1} - t_{k-g}) v_{e,h,k-g} + \dots + (t_k - t_{k-1}) v_{e,h,k-1} + (t_{k+1} - t_k) v_{e,h,k} + (e_i - t_{k+1}) v_{e,h,k+1} = l_i$$

In Figure 7 the delay between $b_{p,1}$ and $e_{p,1}$ has to be described in this way:

$$(t_1 - b_{p,1}) \cdot 1 + (t_2 - t_1) \cdot 1 + (t_3 - t_2) \cdot 1 + (e_{p,1} - t_3) \cdot 1 = l_1$$

Since the multimedia object in the example is not contained in another media object, the environment speed is 1.

To gain the appropriate form of the constraints for the applied solution technique, equations are modified arithmetically so that the right side becomes zero.

Constraints representing resource limits

To guarantee that not more resources are required by the computed schedule than available, resource constraints are introduced. For each different overlapping of media objects in the adaptation interval, we introduce for each resource r constraints of the following form:

$$n_{r,i} v_{i,k} v_{e,i,k} + \dots + n_{r,j} + \dots + w_{r,k} = a_r; \quad w_{r,k} \geq 0;$$

In the equation the constant a_r represents the amount of resource r available in the adaptation interval. A constant $n_{r,i}$ describes the amount of resource r consumed to process one data-unit of the media object i . The variable $w_{r,k}$ is a slack variable which describes how many units of resource r are not used by the presentation.

The terms $n_{r,i} v_{i,k} v_{e,i,k}$ describe how many units of a resource are needed to process a continuous media object i with the speed $v_{i,k} v_{e,i,k}$. The terms have this form, since the amount of a resource which is consumed by a continuous media object depends on its playout speed. The terms $n_{r,j}$ describe how many units of a resource are needed to process a discrete media object j . As discrete media objects have no presentation speed, they have constant resource requirements. Such terms are introduced in the constraint for all media objects that overlap and can share the resource r .

Let us assume that in the example only the resource CPU is considered and that media object 3 and 4 are continuous media objects. There are two different overlapping possibilities in the adaptation interval. Media object 3 can be presented alone and media object 3 and 4 can be presented simultaneously. Hence, we introduce the following two constraints:

$$n_{cpu,3} v_{3,2} v_{1,2} + n_{cpu,4} v_{4,2} v_{1,2} + w_{cpu,3} = a_{cpu}$$

$$n_{cpu,3} v_{3,2} v_{1,2} + w_{cpu,4} = a_{cpu}$$

As can be seen, only the first constraint is necessary because if it is fulfilled the second constraint is also fulfilled. Hence, we can omit the second constraint. In general this means, we can omit constraints the terms describing the needed resources are contained in the same combination in another constraint with additional terms.

Objective function

The objective function of the Mathematical Programme is constructed as follows:

$$\begin{aligned} & (\rho_{2,v,i} - \rho_{1,v,i}) / (\pi_{2,v,i} - \pi_{1,v,i}) v_{i,j} + \dots \\ & (\rho_{2,l,i} - \rho_{1,l,i}) / (\pi_{2,l,i} - \pi_{1,l,i}) l_i + \dots \\ & (\rho_{2,d,i} - \rho_{1,d,i}) / (\pi_{2,d,i} - \pi_{1,d,i}) d_i + \dots \end{aligned}$$

The first row shows how speed variables are represented in the objective function. The second row shows how the variables describing lengths of presentation intervals are represented. The third row shows how variables describing delays implied by interval operators are represented. Variables representing start- and stop-instants are not inserted in the objective function, since only different delays between such instants imply different qualities.

An objective function constructed in this way guarantees that in the optimal solution of the Mathematical Programme the variables $v_{i,j}$, l_i , d_i have values so that the sum of priorities associated with them is as high as possible.

Mapping of arbitrary QoS ranges

If the author has not specified QoS Ranges with two anchor point as assumed above, the following modifications are necessary when the constraints are generated:

- If the author has specified only one single value for an object parameter, we have to insert this value instead of a variable in the constraints.
- If a QoS Range contains three anchor points, it has the form $[\pi_1:\rho_1, \pi_2:\rho_2, \pi_3:\rho_3]$. With such QoS Ranges the following changes have to be made when constraints are generated. We introduce in the constraints instead of a variable representing a speed, a length or a delay the following expression consisting of two variables:

$$\pi_{2,i} - s_i + f_i$$

where the variables s_i and f_i have to fulfill the constraints:

$$0 \leq s_i \leq \pi_2 - \pi_1; \quad 0 \leq f_i \leq \pi_3 - \pi_2$$

In the objective function these variables occur as follows:

$$\dots + (\rho_{3,i} - \rho_{2,i})/(\pi_{3,i} - \pi_{2,i}) f_i - (\rho_{2,i} - \rho_{1,i})/(\pi_{2,i} - \pi_{1,i}) s_i + \dots$$

- If the author has specified more than three anchor points in a QoS Range, it is not possible to consider the whole value range at once. Hence, we have to separate QoS Ranges with three anchor points out of the QoS Range and repeat the whole computation for each such QoS Range. For example, if a QoS Range has the form $[\pi_1:\rho_1, \pi_2:\rho_2, \pi_3:\rho_3, \pi_4:\rho_4]$ we generate the QoS Ranges $[\pi_1:\rho_1, \pi_2:\rho_2, \pi_3:\rho_3]$ and $[\pi_2:\rho_2, \pi_3:\rho_3, \pi_4:\rho_4]$. Then the computation is performed for the first and then for the second QoS Range.

6.3 Computation of the Optimal Solution

A Mathematical Programme generated as described in the last section is solved in two steps:

1. Reduction of the constraints

As not all parameters in the specification will be specified by QoS ranges and as the constraints are generated straight forward from the specification, constraints given as equations can only contain one variable. Hence, equations with only one variable are detected, the value of the variable is determined according to the equation, and the equation is removed from the set of restrictions. If we determine the value of a variable, it is possible that now another equation contains only one variable. Therefore, the procedure is repeated until all equations contain two or more variables.

2. Application of the Generalized Reduced Gradient Method

In general, a Mathematical Programme generated as described in the last section is not linear. The objective function is always linear, but restrictions integrating speed variables are not linear. Hence, we have to apply a solution technique for non-linear Mathematical Programmes. We have tested the application of a Gradient Projection Method [7] and an Exterior Penalty Method [7], but the Generalized Reduced Gradient Method (GRG) [8] showed the best performance.

The GRG method is a solution technique for non-linear Mathematical Programmes, where the optimal solution is iteratively computed starting from a valid solution of a Mathematical Programme. A valid solution is a solution which fulfills the constraints of the problem but does not necessarily

maximize the objective function. A valid start solution is computed also by the GRG method as described in [8].

The complexity of one iteration step of the GRG-method is in the worst case $O(m^3)$, where m is the number of constraints. Because of the special structure of some of our constraints, optimizations are possible that reduce the average complexity of one iteration:

- The GRG-Method can be modified, so that variable boundaries can be considered by the algorithm and need not be described as constraints. Hence, the number of constraints can be reduced significantly and we have only to consider constraints of the form $g_i(x) = 0$ explicitly.
- The expensive computation task of the GRG-Method which causes the complexity of $O(m^3)$ is the inversion of the Jacobian of the constraint functions $g_i(x)$ at the current solution x^i . This is the matrix $\partial g / \partial x (x^i)$. Because of the special structure of some of our constraints, it is possible to speed up the computation of inverse matrixes as follows: The constraints $g_i(x)$ are ordered such that constraints, which contain a variable that is not contained in another constraint, obtain low numbers i . Then, the variables are ordered such that variables, which appear only in one constraint, obtain the same number as the constraint in which they are contained. The result of this ordering is that the first n columns of the Jacobian-Matrix have only values in the diagonal, where n is the number of variables which appear only in one constraint. To invert such a matrix, we use the LU decomposition method [9] which computes the inverse matrix column by column from the original matrix. A corresponding column of the inverse matrix of a column of the original matrix, which has only a non zero value in the diagonal — according to its position in the original matrix —, is a column with the reciprocal of the non zero value at the same position. Hence, the first n columns of the inverse matrix can be determined in $O(n)$ steps. For the remaining columns $O(m^2 * (m-n))$ steps are needed according to the LU decomposition method. Thus the complexity of the matrix inversion is still $O(m^3)$, but in average the computation will be much faster.

7. Updating of Scheduling Graphs

When the presentation has been performed according to the computed values in the adaptation interval, the event ranges of nodes, which lie in the adaptation interval, are set to the computed instants. By doing this for all scheduling graphs, the course of the presentation is documented.

Different scheduling graphs represent various presentation alternatives. Thus, it might happen that scheduling graphs, which are currently not used to control the presentation, contain additional start nodes of presentation intervals in the adaptation interval. If such a node belongs to a presentation alternative of a selection group for which the selection policy ‘static’ or ‘dynamic’ was specified for, we delete all instants from the event range of the node that lie before t_{k+1} . If the event range becomes empty, it is no longer necessary to consider that scheduling graph because the presentation alternative represented by the scheduling graph can not be implemented any more. If such a node belongs to a presentation alternative of a selection group the selection policy ‘init’ was specified for, we can delete that scheduling graph because in further renditions of the scene we have always to implement the presentation alternative which was once selected.

If the event ranges of some nodes in a scheduling graph are modified, the event ranges of nodes which depend on them have also to be updated. Figure 9 shows a scenario where a part of a media object has already been presented. Hence, the event range of node $e_{p,3}$ computed so far as

$$N(e_{p,3}) = N(b_{p,3}) + l_3 / v_{e,3}$$

is no longer valid. Because now it is only allowed that $e_{p,3}$ occurs at the instants

$$N(e_{p,3}) = t_k + (l_3 - (t_k - b_{p,3}) v_{e,3,k-1}) / v_{e,3,k}$$

where $v_{e,3,k}$ is the value range of all speed values possible for the media object behind t_k .

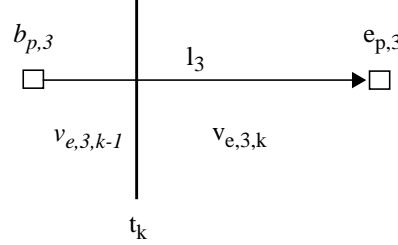


Figure 9: Update of event ranges

This means, we have to update the event ranges of nodes in each scheduling graph according to this schema where an edge ends which starts at a node that lies before t_k . Then, we can compute the event ranges of the other nodes by applying step 2 and 3 of the event range computation algorithm.

8. Related Work

Various temporal models and synchronization concepts have been developed for multimedia presentations. Multiple variations of Petri Net models such as OCPN [6] or TSPN [10] exist which have a strong expressive power according to the specification of synchronization aspects. However, these models do not adapt documents with regard to the resource situation.

Besides Petri Net-based models, several high-level temporal models have been proposed. Models that base on the time-line concept, such as MAEestro [4], integrate no flexibility. Hence, it is not possible to compose adaptable specifications. Some models that describe temporal relations between media objects explicitly, support indeterminism. E.g., Mbuild [5] is based on an extension of “Glue” known from T_EX. Temporal Glue relates media objects in a flexible manner. In FLIPS [11] temporal relations are defined flexible by barriers and enablers. Barriers prevent a media object from being presented or stopped until another media object has stopped or started and enablers cause a media object to be presented or stopped when another media object has stopped or started. In these models the contained indeterminism is not applied to adapt the specification, it helps either to ease the composition of presentations or to synchronize presentations where interaction is possible.

Firefly [1, 2] integrates mechanisms to specify flexible presentations. Firefly is event-based, which means that the start and end of media objects are modeled as instants. The delay between related events is described by a minimum, optimal and maximum value. Additionally, costs are defined for shrinking the extent of an object to minimum or extending it to maximum. To automatically compute the optimal schedule out of these values, a linear programming algorithm has been proposed. To gain adaptable specifications, the cost values could be defined with regard to the relevance or resource requirements of media objects. However, the authors of Firefly do not exactly define how cost values are determined and the presentation system does not consider the resource situation when the presentation schedule is created.

In CHIMP [3] temporal aspects of media objects are modeled by flexible constraints, which allow to specify ranges for the temporal values. It is possible to specify alternative constraints and to assign priorities. The concept to adapt presentations to different resource situations is equivalent to the concept we propose. It is also tried to distribute the available resources between simultaneous presented media objects on the basis of resource information. Compared with CHIMP, our model allows a finer granularity

with regard to the specification of priorities. Further, CHIMP does not integrate abstractions such as selection groups.

9. Conclusion

To perform multimedia presentations under different resource situations, adaptable documents models are required. We have identified various forms of flexibility which make multimedia documents temporally adaptable. The presented Tiempo model offers selection groups to represent alternative presentation parts consisting of media objects and interval operators. With QoS ranges alternative presentation options of media objects or interval operators can be defined. Assigned priorities define which alternatives of selection groups or QoS ranges are preferred by the author. Hence, Tiempo conform multimedia documents can integrate a high degree of flexibility and resource shortages need not result in a reduced presentation quality.

The presented adaptive scheduling algorithm allows to adapt flexible presentation to changing resource situations. It updates the presentation schedule at regular intervals so that the optimal presentation quality is implemented in the current resource situation. This is achieved by determining which presentation parts or presentation options have to be implemented so that the available resources are distributed optimal between simultaneously presented media objects.

Based on the described concepts, we currently develop a presentation system for system environments with best-effort assignment of resources.

10. References

- [1] M. Cecilia Buchanan and Polle T. Zellweger. "Scheduling Multimedia Documents Using Temporal Constraints", in Proc. of the 3rd International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, USA, pp. 223–235, 1992.
- [2] M. Cecilia Buchanan and Polle T. Zellweger. "Automatic Temporal Layout Mechanisms", in Proc. of ACM 1st Intl. Conference on Multimedia, Anaheim, USA, pp. 341 – 350, 1993.
- [3] K. Selcuk Candan, B. Prabhakaran and V.S. Subrahmania. "CHIMP: A Framework for Supporting Distributed Multimedia Document Authoring and Presentation", in Proc. ACM Intl. Conference on Multimedia, Boston, USA, pp. 329-339, 1996.
- [4] George D. Drapeau. "Synchronization in the MAEStro Multimedia Authoring Environment", in Proc. of ACM 1st Intl. Conference on Multimedia, Anaheim, USA, pp. 331 – 340, 1993.
- [5] Rei Hamakwa, Hidekazu Sakagami, and Jun Rekimoto. "Mbuild - Multimedia Data Builder with Box and Glue", in Proc. of IEEE 1st Intl. Conference on Multimedia Computing and Systems, Boston, USA, pp. 520–525, 1994.
- [6] T.D.C. Little and A. Ghafor. "Synchronization and Storage Models for Multimedia Objects", IEEE Journal on Selected Areas in Communication, vol. 8, no. 3, pp. 413-427, 1990.
- [7] M. Minoux. "Mathematical Programming - Theory and Algorithms", John Wiley and Sons. 1986.
- [8] Klaus Neumann. "Operations Research Verfahren, Band I", Carl Hanser Verlag, München, Wien, 1977. (in German)
- [9] Press, Flannery, Teukolsky, Vetterling. "Numerical Recipes - The Art of Scientific Computing", Cambridge University Press. 1986.
- [10] P. Senac, Michel Diaz, Alain Leger, and Pierre de Saqui-Sannes. "Modeling Logical and Temporal Synchronization in Hypermedia Systems", IEEE Journal on Selected Areas in Communications, vol. 14, no. 1, pp. 84-104, 1996

- [11] J. Schnepf, J. A. Konstan, and D. H.-C. Du. "Doing FLIPS: FLeXible Interactive Presentation Synchronization," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 1, pp. 114-125, 1996.
- [12] Thomas Wahl and Kurt Rothermel. "Representing Time in Multimedia Systems," in *Proc. of IEEE 1st Intl. Conference on Multimedia Computing and Systems*, Boston, USA, pp. 538-543, 1994.
- [13] Thomas Wahl, Stefan Wirag and Kurt Rothermel. "TIEMPO: Temporal Modeling and Authoring of Interactive Multimedia", In *Proc. of IEEE 2nd Intl. Conference on Multimedia Computing and Systems*, pp. 274-277, Washington DC, 5 1995
- [14] Stefan Wirag, Kurt Rothermel and Thomas Wahl. "Modeling Interaction with HyTime", In *Proc. GI/ITG Kommunikation in Verteilten Systemen*, Germany, Chemnitz, pp. 188-202, 2 1995