Document Model Issues for Hypermedia

Lynda Hardman and Dick C.A. Bulterman
Interoperable Multimedia Systems Group
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Email: {Lynda.Hardman, Dick.Bulterman}@cwi.nl

A number of different systems exist for creating multimedia or hypermedia applications—each with its own internal document model. This leads to problems in comparing documents created by these systems, and in describing the information captured by a document for long-term (system independent) storage and future playback.

We discuss the components which should be considered for a hypermedia document model. These include the hierarchical and linking structure of a document and the spatial and temporal relations among components of the document. Other aspects, such as transportability of documents and information retrieval, are also addressed briefly.

We present the Amsterdam Hypermedia Model which, while expressing only a subset of all possible structures, has been used as a basis for a comprehensive authoring environment.

1. Introduction

Although hypermedia is often thought of as something innovative, it has been developed to make explicit already existing, but implicit, relations among pieces of information. A hypermedia model can be used to describe interactive aspects of familiar communication media. A television news program, for example, can be described in terms of a hypermedia presentation—initially there is an introduction by a newscaster; this leads into a film clip, normally accompanied by some commentary, on a particular news story; then we see the newscaster again. In this case the user is not making any choice, but the action of "jumping" to a new scene is present. By extending this example only slightly, a hypermedia presentation can be made by playing the same introduction, then giving the user the choice of which film clips to see, then returning to the newscaster for a further selection. A more static example of interaction is a book, or paper, where a reader can take notes. Sections of the text can be marked as relevant, and notes can be written in the margin either agreeing or disagreeing with the author. A future reader is able to see which notes are attached to which pieces of text.

Both examples show relations among media items (atomic pieces of multimedia data) which we would like to be able to preserve in on-line, interactive presentations. In order to be able to express these, and to enable information to be captured for later use, an information model is required in order to formally define the structure of the information, to efficiently map the structure to storage, and to support information and

The Handbook of Multimedia Information Management.
Ed.s W.I. Grosky, R. Jain and R. Mehrotra. Prentice Hall 1997.

retrieval [7]. A model can also be used to compare the expressive power of hypermedia authoring systems, and for transporting presentations from one platform to another.

A requirement for a useful hypermedia model is that it can describe sufficient complexity so that the essence of a presentation can be preserved from one platform to another. This includes specifications of the media items used, the temporal relations among the items, layout of the items and interaction possibilities with the presentation. On the other hand, when a model becomes more complex
there is a danger that it becomes too difficult to specify for any particular presentation, with consequences that an authoring system becomes very complex to use. In the extreme case a hypermedia presentation can be programmed directly in a non-specialist programming language, giving flexibility but minimal reuse. A simple model, supported by easy-to-use tools, is in turn too restrictive to allow the creation of anything more than, say, the sequential presentation of a slide show. The creation of a useful model is to find a pragmatic trade-off between these two extremes.

We present the different aspects of hypermedia modelling in this chapter through the use of an intuitive three-dimensional representation of a multimedia presentation, for example see Fig. 3. This consists of the notion that media items are displayed on a screen for some duration with timing and layout relations specified among them. A solely time-based model such as this is insufficient, since at some, normally unpredictable, time the user may interact with choice points and jump to another part of the presentation. On the other hand, a model based only on the hypertext notion of interaction, [15], is also insufficient since it lacks the timing constraints among the media items. Often the term hypermedia is used to describe linked, multiple media, where time is only relevant when playing a particular media item and not for the presentation as a whole. We take a more multimedia based approach, that time needs to be integrated into the document model.

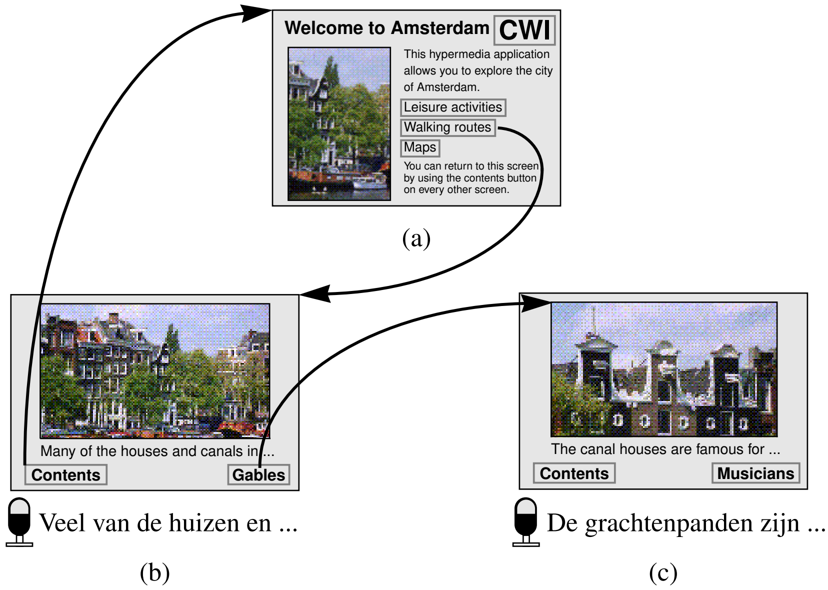
As well as a model satisfying the direct issues of recreating a presentation in a different environment, a hypermedia model also has to take into account other issues, for example information storage and retrieval. While the main purpose of a hypermedia model is not to satisfy these sorts of requirements, it should remain compatible with them.

In this chapter we do not address authoring issues of hypermedia, these are discussed at length in [18]. Nor do we address issues of different data types or storage models for data, nor playback issues of a presentation, such as synchronization of remote sources (see the Baqai and Ghafoor chapter in this volume). We do discuss briefly a number of languages for expressing a hypermedia model (section 3.5) but this is not a main theme of the work.

We begin our discussion on hypermedia documents models with an example of a hypermedia presentation, to give a base on which to discuss the abstractions in the model. We then discuss a number of issues that require to be addressed in a hypermedia model, with reference to other systems and models.

2. An Example Hypermedia Presentation

As a starting point for our discussion on models, we consider the characteristics of a "typical" hypermedia presentation. Hypermedia presentations vary widely in terms of their interactivity. For example, an entertainment application such as watching a video



(a) Contents screen of the Amsterdam tour. Four media items (3 text, 1video) are displayed. There are three interaction choices from the long text item and one choice from the CWI item. Selecting the *Walking routes* option takes the user to (b).
(b) The first scene in the walking route. Three text items and one image are displayed; a fifth item is a Dutch spoken commentary. Selecting *Contents* takes the reader to the contents screen in (a); *Gables* takes the reader to the scene in (c).
(c) The gables scene in the walking route, similar in composition to (b).

Figure 1. An example interactive multimedia presentation.

is comparatively passive for the user. A video game, in contrast, is a highly interactive activity for the user. An activity falling between these two extremes might be the reading of a (hypermedia) newspaper, where the user can choose which news items to watch, and sit passively while an item (perhaps video, perhaps text) is presented. Both the more passive and the more interactive ends of this interaction spectrum demand the transmission of data and the synchronization of the constituent media items. Often the term multimedia is used to describe more passive presentations, where interaction is deemed of lesser importance than the (pre-specified) interactions among the media items.

An example of a “medium interactivity” presentation is shown in Fig. 1, which illustrates three fragments from a tour of the city of Amsterdam. In the top fragment, which is analogous to a table of contents, a user is given a description of the tour and a number of choices that can be selected within the presentation. One of these choices is illustrated—containing a description of walks through the city, highlighting several features found on the tour, which is itself sub-divided across a number of other fragments. From a media perspective, each fragment consists of a number of media items displayed for some duration on screen or played through loudspeakers. This structure is shown from a time perspective in Fig. 2. Here, a number of items are presented simultaneously, then when the user makes a choice to jump to the walking route, the cur-

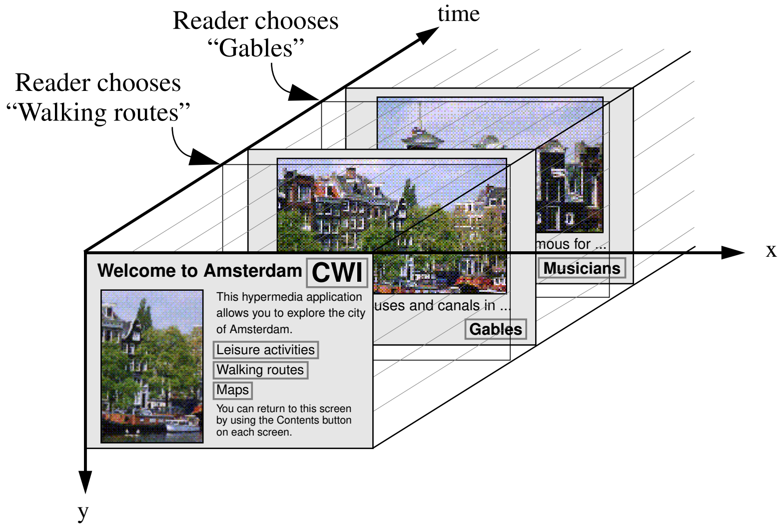


Figure 2. Time-based view of a possible route through Fig. 1.

rently running presentation is left and a new presentation is started. Here a number of items are again started simultaneously, but after a few seconds the subtitle changes to keep in time with the spoken commentary. These time relations are an example of the types of information that need to be captured by a hypermedia document model.

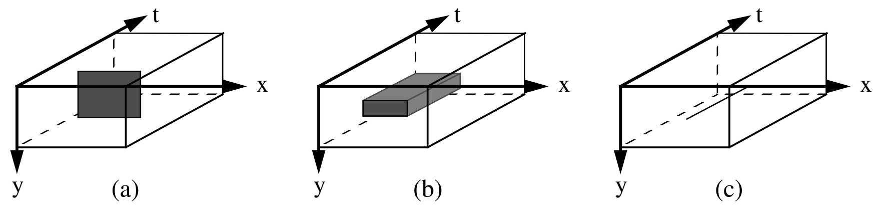
3. Model Issues

The issues that need to be addressed in a hypermedia model can be broadly categorised into document specification, document transportation and information representation. Document specification is of the main importance in this work, where we are interested in specifying sufficient information for recreating the hypermedia presentation on a number of heterogeneous platforms, and for ensuring that future systems will also be able to interpret the information. We address these issues in the four sections: Media items, Structure, Presentation (including timing and layout relations), and Interaction. Structural relationships define logical connections among items, including the grouping of items to be displayed together and the specification of links among these groupings. Layout specifications state where screen-based media are to be sized and placed, either in relation to each other or to the presentation as a whole. Timing relations specify the temporal dependencies among media items, possibly stored at different sites. Interactions, including navigation, give the end-user the choice of jumping to related information.

Document transportation, although not within the scope of the model, influences the translation of the model to some language expressing the model. This is discussed briefly and then followed by some observations on the storage and retrieval of information and their implications for requirements of a document model.

While we prefer to use the term hypermedia to refer to collections of multimedia presentations through which the user can navigate, we will sometimes return to the term multimedia. We use the term *multimedia presentation* to indicate a collection of media items related temporally and/or spatially, or, in other words whose presentation is defined in terms of the same space and time coordinates. We show below, section 3.2.1 and section 3.2.3, that a hypermedia presentation can be composed of not only a single multimedia presentation, but of multiple independent multimedia presentations playing simultaneously.

3.1. Media items



(a) Text or graphics—spatial but no intrinsic temporal dimension.
(b) Video or animation—spatial and temporal dimensions.
(c) Sound—temporal but no spatial dimension.

Figure 3. Spatio-temporal dimensions of media types

It is useful to discuss the properties of the media items making up a presentation before going into the more complex aspects of document modelling. A media item may consist of a single monomedium, such as text or sound, or a composite medium such as interleaved video and sound. In either case, we use the term *media item* to refer to a data object used as part of a presentation.

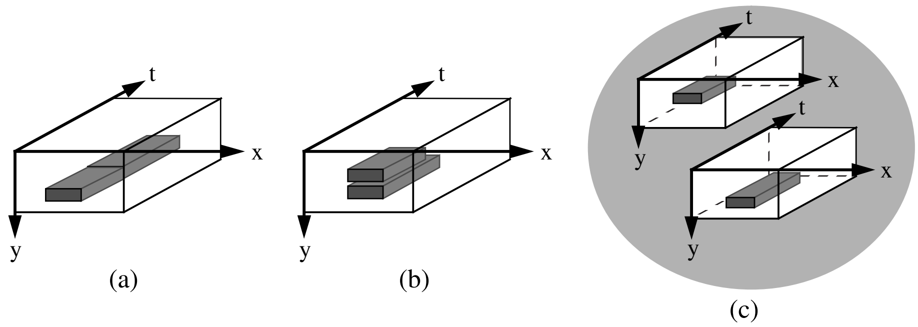
Media items have data-type dependent properties, the most important of which for multimedia is their relation to space and time. Fig. 3 shows representations of these in a three-dimensional space. Text, an ordering of letters, requires two dimensions for display. The aspect ratio of the display area is relatively unimportant, since words can be wrapped without losing the meaning of the sentences. A time dimension can also be applied to text, for example the time it takes for a user to read the text. This timing, however, is not part of the data structure of the object. Images also require two space dimensions for display, but aspect ratio is important, since real-world representations of objects can be distorted, not to mention aesthetic considerations. (We treat image items such as pixel-based images and vector graphic images as being fundamentally the same in this spatial view of the data. The editor CMIFed [32], for example, treats a number of different image data types as the same type of object.) For both text and image media items a duration can be assigned by an author. Video requires all three dimensions of space and time to be displayed, and can be regarded as a sequence of images, each image being displayed at a particular time. As is the case for an image the aspect ratio is important, and the rate of display of frames should also remain constant for faithful reproduction. We regard animations (normally vector-based graphics) as having properties similar to video. The final media type, sound, has one time dimension and no spatial dimensions.

A media item does not necessarily have to be a complete file or object. It might be a reference to part of a physically stored object. The media item is then a combination of a reference to the stored data item and a (media-dependent) specification of the part of the item to be played. In the case of text the object may be a complete book, where only a section is required. For an image, a portion of the image is cropped. In sound, for example, a selection from a music item may last a number of seconds, but may also be only one track for the length of the complete item. A video segment might be a combination of temporal and spatial cropping operations, where a number of frames are selected from the complete sequence (cropping in time) and only a part of the image is shown (cropping in space).

Other media item types that may be included in a presentation are outputs from external processes—for example a video camera pointing at a scene outside, or the reading from a monitoring device in a chemical plant or power station. We can still treat it as an object of similar space dimensions (video in the first example, a text string or image in the second), but then with an unknown or indefinite time dimension. Alternatively, it may be a media item of known type and duration, but generated on-the-fly from an external program. For example, financial results are generated from a market simulation program and displayed as an image in the presentation, [20].

3.2. Structure

3.2.1. Composition



(a) Time-dependent composition, serial.
(b) Time-dependent composition, parallel.
(c) Time-independent composition.

Figure 4. Composition

A hypermedia presentation can be regarded as a hierarchical structure with media items at the leaf nodes, [21]. This allows groups of items to be treated as one object. The MET^{++} system [1], for example, allows an author to group media items into a composite object and then manipulate the composite by, e.g., increasing the time duration of the composite and distributing the changes among the constituent children. This is analogous to a graphics editor which allows grouping of diagram elements and stretching/shrinking of the group as a whole.

Although a number of multimedia authoring systems allow parallel and serial composition ([1], [16], [23]) and treat these as fundamental divisions, they are in fact two extremes of one form of composition—*time-dependent composition*. This is where two, or more, items (or groups of items) are grouped together in one composite and a time relation is specified among them. In the serial case, Fig. 4(a), the time relation is that one starts when the other finishes; in the parallel case, Fig. 4(b), that the items start together. An intermediate case is that one item starts and at some time later, but before the other finishes, the second one starts. (We continue the discussion of timing relations in section 3.3.2.) Normally items are displayed on the screen without any thought of spatial relations, although mostly they are implicit, for example subtitles appear at the bottom of the video they apply to. The determining factor is that the items are being grouped in the same space/timeline. Fig. 4 is perhaps misleading, since it implies that serial composition requires that the two objects occupy the same screen position. The only condition that may apply is that two items in parallel cannot occupy the same position (unless the author explicitly specifies overlap).

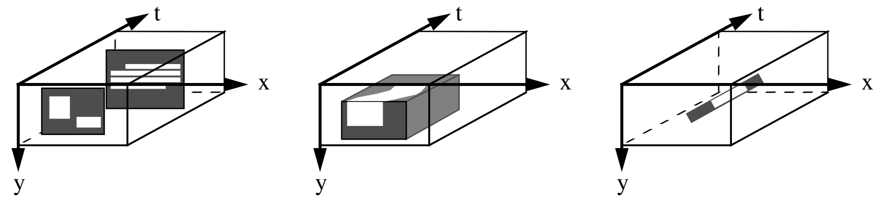
The other form of composition is *time-independent composition*. This allows the grouping of items that have no time or spatial relations with each other, Fig. 4(c). They *may* be played at the same time, but there is *no pre-defined* relation between them. The utility of this is perhaps not at first clear, but it enables a number of situations to be modelled. For example, two or more presentations may be played at the same time: in Fig. 1 if the reader selects the CWI logo, then a spoken commentary is given about the institute. The timing of the spoken commentary is not bound to the other media items already on the screen, but is conditional on the reader selecting the logo. This "unpredictable" behaviour can be modelled by explicitly separating the time bases of the two presentations. Another example is where a presentation is built up of several subscenes. A number of items remain playing on the screen (e.g. a heading, background music, a link back to a contents screen) while the reader selects the different subscenes (e.g. a picture with spoken commentary)—again, there is no timing relation between the items that remain on the screen and those playing in the subscenes. In each of these examples, the timelines of the two presentations that are playing are independent. The spatial relations between the two presentation are also independent (there is no explicit constraint), but when playing the presentations they should not, normally, occupy the same screen position. A practical way of ensuring this is to use separate windows for each of the presentations in the first example, or to restrict playing of a subscene to a specific subwindow in the second example.

When the time-dependent compositions serial and parallel are supported in an authoring system, these can be used to calculate timing constraints for the presentation, as is the case in Mbuild [16] and CMIFed [23].

3.2.2. Anchors

Anchors were introduced in the Dexter model [15] as a means of referring to part of an item in a presentation in a media-independent way: they are an abstraction which allows data-dependencies to be kept hidden. An anchor value specifies a part of a media item (there are other aspects to an anchor object which we will discuss as part of the model in section 4.1). The main use for anchors is to provide a source or destination object for linking within and among presentations, when they are used in conjunction with links (see section 3.2.3). Another use is to provide a base on which to attach temporal relations so that parts of media items can be synchronized with one another (see section 3.3.2), or even as a base for spatial relations (see section 3.3.1).

In Fig. 5 a graphical interpretation is given for text, image, video and sound anchor values. A text anchor normally specifies a sequence of characters within a text item. An



(a) A text anchor is likely to be a text string, a graphics anchor an area.
(b) A video anchor is ideally an area changing with time.
(c) A sound anchor is a temporal extent, e.g. in music a temporal extent within an instrument.

Figure 5. Anchor values

image anchor specifies an area in a pixel-based image, where most systems implement the area as rectangular, although there need be no restriction on the shape of the area and any contour[1] could be defined to specify the extent of the area. In a vector graphic image an anchor may refer to any object (single or grouped) in the image—the point is that the internal specification of the anchor value is data format dependent.

Within a video item an anchor may be chosen as a sequence of frames, as is used in a number of systems [12], [24]. This allows the user to select at most one link to follow to another presentation at any frame in the video. A more desirable approach is to specify the area on the screen for the extent of the frame sequence [35]. This allows several choices for each frame, but the choices do not vary during the sequence. A complete description is given when each area is specified per frame in the sequence, [10]. This allows moving or changing objects in the video to be followed, so that clicking on an object becomes a more natural option. This last description is illustrated in Fig. 5(b).

For sound items it is intuitive to describe the sound anchor, e.g. it is a stretch of sound (illustrated in Fig. 5(c)), or it might be a stretch of sound in one sound channel, for example a number of bars of violin solo. The problem starts when the reader tries to interact with the sound item, since with the normal mode of interaction with hypermedia presentations (clicking an object on the screen) there is nothing tangible with which to interact— although links *to* sound items remain possible. An example of interacting with "hyperspeech" is given in [3].

Anchors are not restricted to identifying parts of single media items, but can also be used to specify parts of groups of items. For example, in Fig. 1 (b) the words "houses" might be a text anchor, one of the houses in the picture an image anchor, and the Dutch word "huizen" part of a sound anchor. All three anchors may be combined to form a composite "house" anchor which can be referred to from other parts of the presentation.

As well as defining the extent of part of a media item via the anchor value, an anchor can also have attributes associated with it. These can be used for labelling the parts of the media item they refer to [10]. This labelling of anchors becomes a route into attaching information to media items which can be used for retrieval of items—we discuss this further in section 3.6.

1. It need not even be a connected contour.

3.2.3. Links

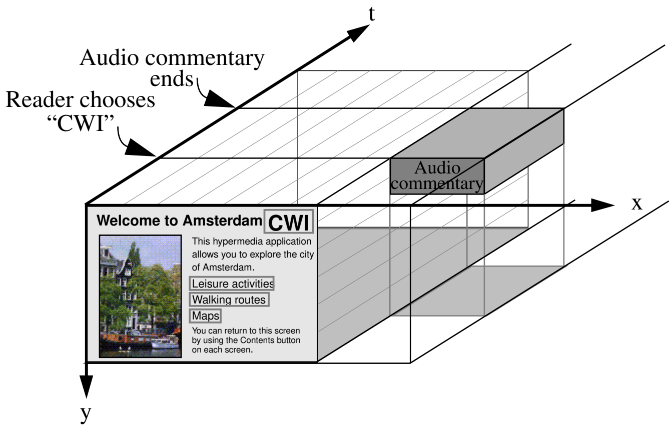
Links are defined as part of the Dexter hypertext model [15] for explicitly representing relations among objects. They specify a logical connection between two (or more) end points, specified via anchors. Most hypertext systems allow a user to follow a link as the basic form of interaction with the document structure. The use of links in hypermedia similarly allows the user to make choices as to which presentations to view and captures this in the document structure. The problem with links in multimedia is that a presentation normally consists of a number of media items playing simultaneously, and any one of these may have its own duration. In other words, links are not from static text or image items, as is generally the case in hypertext, but from a complete multimedia presentation. This leads to the question of where links fit into this more dynamic and complex document structure.

In the Dexter model a link has source items and destination items. Systems supporting the model do not often use composition of items, so that the structure is often fairly flat—one media item is displayed, a link is followed and a new item is displayed. In the multimedia case, where the presentation is invariably composed of a number of media items the question is how many of the items are associated with each end of the link. For example, in Fig. 1, following the link from *Walking routes* in (a) to the screen in (b) results in the complete window being cleared and the new presentation being displayed. In the case of following the link from *Gables* in (b) to the scene in (c) all the items except the *Contents* text are cleared. In this case, the scope of the information associated with the link is only a part of the original presentation. We call this scope specification a *context* [22].

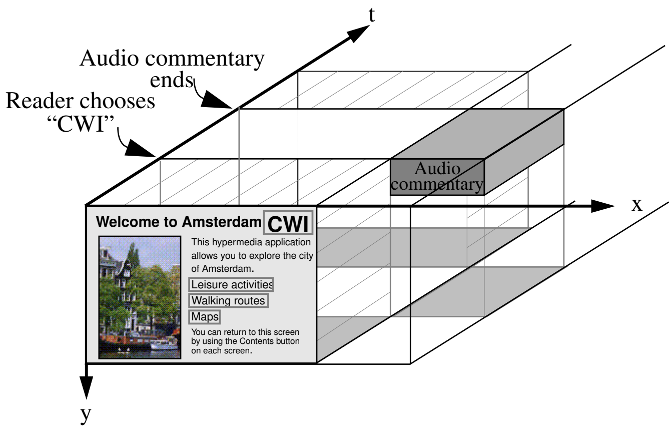
A consequence of associating a context with sources and destinations of links, is that the same anchor can be used in different links with different contexts. An example is when following a link from an anchor, say *Gables* in Fig. 1(b), not all of the presentation is cleared (in this case everything except the *Contents* item). Following a link to the same anchor with the whole scene as destination context, however, would result in playing *all* the items in (b).

Presentation specifications can also be associated with a link. These become more varied in hypermedia. When following a link in hypertext, systems generally display the destination items of the link, and either remove or leave the source items. The problem is that it is not specified in the model what the action should be, but is left to the particular system to interpret the action in its own way. The required action on following a link can be specified in a model by recording what happens to the link's source and destination contexts on following the link. For example, in Fig. 5 (a) a presentation is playing and when the user follows the link the new presentation is played, while the original presentation continues playing. In Fig. 5(b) the situation is similar, but in this case the original presentation pauses while the new presentation plays. The only difference between these two cases is brought about by the fact that the original presentation can pause or continue playing. The third case, in Fig. 5(c) is where the original presentation is removed and the new presentation is played on its own.

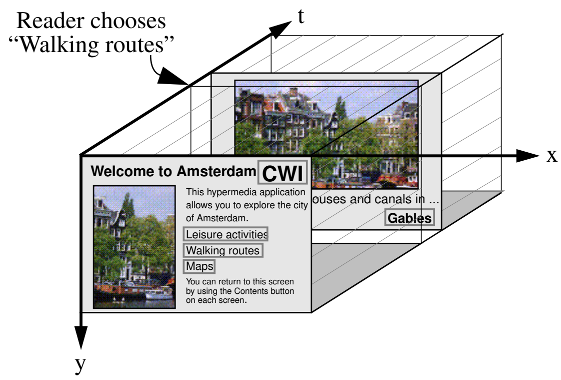
Other presentation specifications can be associated with a link, for example it might specify that the anchor in the destination context blink to make it more visible. A further presentation feature, normally found in multimedia systems, is the transition [34]. Here, for example, when an image is replaced by a new image there is a choice of a number of actions such as fading the original image out and fading the new image in;



(a) New presentation plays in addition to original presentation.



(b) Original presentation pauses while new presentation plays.



(c) New presentation replaces original presentation.

Figure 6. Links in hypermedia

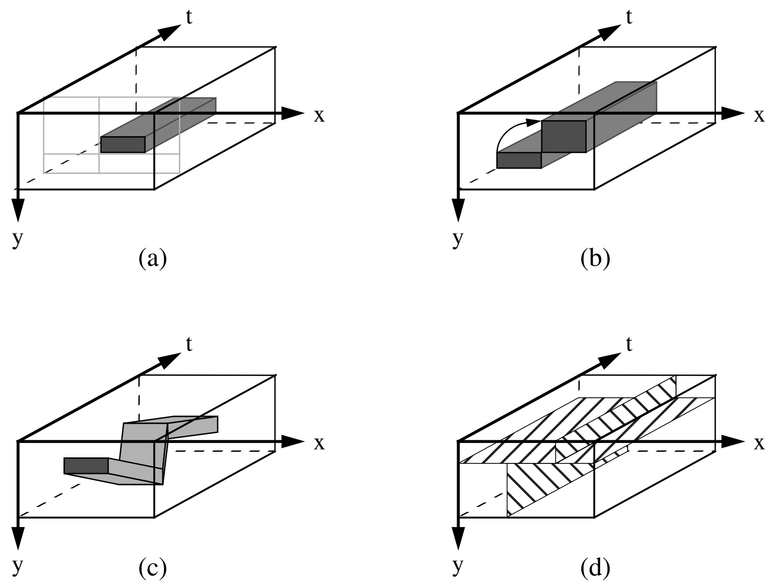
similarly videos can dissolve into one another. Transitions can also be specified as a presentation attribute of the link when the destination context of a link is played in the same screen area as the source context. The transition might even be a sequence in its own right. For example, if a user chooses to zoom in to Amsterdam from a map of the earth, actioning the link doesn't make the presentation jump to a presentation about Amsterdam, but increases the scale of the earth gradually then dissolves into the new presentation.

Links specify relations among objects, and, when labelled with semantically relevant attributes, can be used for building up information structures. We discuss this further in section 3.6.

3.3. Presentation

3.3.1. Spatial Layout

Spatial layout specifications of items can be given in a number of ways. The most common method is to define an item's position per item and relative to the window (or screen) in which it will be displayed. Examples of authoring systems using this approach, illustrated in Fig. 7(a), are Eventor [14], Mbuild [16], Authorware and Director [36]. Just as valid a method of specification, but to our knowledge not used in multimedia authoring systems, is to define the coordinates of a media item relative to some other media item, illustrated in Fig. 7(b), or relative to an anchor within an item. This would be useful for displaying, for example, subtitles next to the video they



(a) Position defined relative to window.
(b) Position defined relative to another item.
(c) Object moves as a function of time.
(d) Channels predefine screen or window areas.

Figure 7. Spatial layout.

belong to, since if the position of the video is changed, the subtitles will remain in the same neighbourhood relative to the video. (We, and others, make similar arguments for relative timing constraints in section 3.3.2.) While it is not (yet) common to specify position relative to other items, some systems do allow the movement of objects with time, for example the MET^{++} [1] and Eventor [14] systems. This is illustrated in Fig. 7(c).

While defining the position of each item explicitly is relatively common, for large presentations it can become difficult to maintain an overview of the different layouts used. An approach to solving this is implemented in CMIFed [23], where layout objects (called channels) are defined which predefine areas of the window into which media items can be played. This is illustrated in Fig. 7(d). This allows an author to make changes to a channel which then apply to all the items played in the channel. While the current implementation plays an item in the middle of a channel, there is no reason why the position of a media item should not be specified relative to a channel.

A combination of the methods mentioned could be made, where a channel can move in time, another channel can be defined relative to the first, and an item's position within a channel can change with time. While this does not perhaps seem immediately useful, it should be included in an encompassing hypermedia model.

Channels, as used in CMIFed, are also used for defining high-level presentation specifications. These may be media-independent, for example background colour, or media dependent, for example font style and size. This is again useful for making global presentation changes to the presentation. This high-level presentation specification is used as a default, and can be overridden by specific layout specifications from individual media items.

The channels define areas relative to the window, so that in an environment where the window size can be changed, the channels also change in proportion. This means that a presentation is not defined for a fixed window size. The aspect ratio of the window may also change. In the current implementation images are scaled to fit into the available area—preserving the image's aspect ratio. (Cropping, as described in section 3.1, could also be used.) A font scaling algorithm would also be useful for preserving the "look and feel" of the overall presentation.

While layout is, in theory, independent of the logical structure of a presentation, it is clear that presentations created from different levels of structure need to have coordination of layout among the different items. This resolution is not specified in the model, but is left to the system authoring the presentation. For example, CMIFed restricts the playing of media items in a channel to only one at a time.

3.3.2. Temporal Layout

Temporal layout is the determining characteristic of multimedia, in the same way that links form the basis of hypertext. In this sense, temporal properties take on a much greater importance than a small section of a hypermedia model. Our intention, however, is to make explicit where these temporal relations fit in with other relations in such a model, and give an overview of the types of temporal information that can be specified. We do not discuss, for example, how these relations can be encoded, nor how a system can strive to execute the specified relations. The issues discussed in this section are derived from the work described in [6] and [13].

Timelines

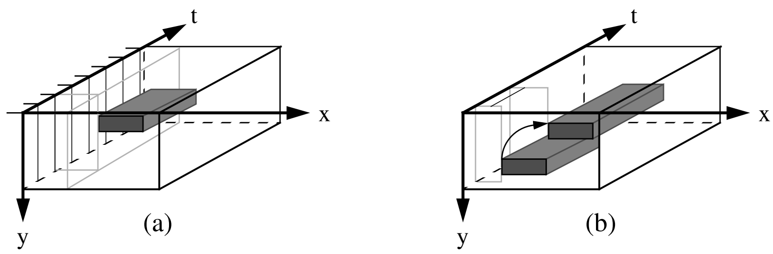
When specifying when a media item should be played in a presentation there are two different ways of providing this information. The first is to specify when the media item should start in relation to a timeline, Fig. 8(a), the second by specifying the relation with respect to another object in the presentation, e.g. another media item Fig. 8(b) or a group of objects. Both cases are supported by a number of systems: the former approach, for example, by Director [36] and the Integrator [34]; the latter, for example, by Firefly [5] and Eventor [14]. Each approach has advantages and disadvantages. The use of a timeline makes it possible to make changes to the timing of one object without affecting other objects in the presentation. This, however, requires the re-specification of all items if, e.g., one item is removed and the others have to be brought forward to fill in the gap. Specifying relations among objects, in contrast, provides more flexibility when editing the presentation later, but requires that all the relations be specified.

When defining temporal relations among the objects themselves, there remains an implicit timeline, if only the “real-time” timeline that is moved along when the presentation is being played to the user. In both cases (implicit and explicit timelines) the rate of traversal along the timeline can be varied, in a similar way that music can change its tempo when being performed. A number of systems use the timing relations specified among objects to generate a timeline, [14], [16], [23].

Durations, points in time and intervals

The duration of a media item may be explicit or implicit, that is derived from relations with other objects. For example, a video has an explicit time associated with its data (the number of frames in the clip divided by the frame rate), or an image is given an explicit display duration. An example of a derived relation is where a subtitle starts when a spoken commentary begins and remains on display until the commentary has finished. In order to achieve this type of derived duration, media items need to have the property that they can be scaled. (We consider the properties of scaling in section 3.3.4.)

A part of a media item may be a point in time (e.g. a frame number in a video, or “5 seconds after the start” of a sound item, or “the beginning of the word *gables* in a spoken commentary), or an interval (e.g. a number of frames in a video, “between 3 and 5 seconds” in a sound item, or the duration of the word *gables* in a spoken commentary). An interval may be specified by a combination of points or intervals (e.g. “from the beginning of the word *distinctive* to the end of *gables* in a spoken commentary”).



(a) Time specified with respect to a timeline.
(b) Time specified with respect to another media item.

Figure 8. Temporal layout.

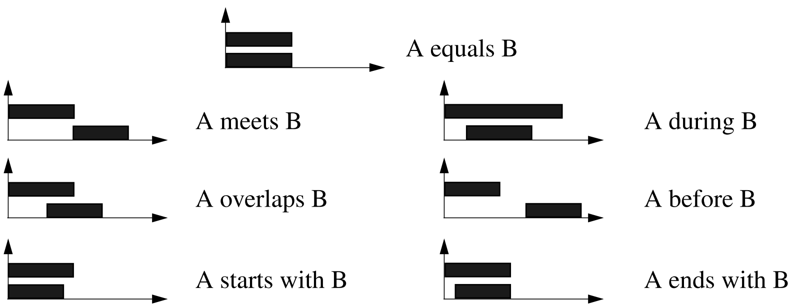
Temporal relations
Temporal relations among objects can be defined in terms of whole media items, parts of media items, or groupings of items. Examples are: (a) a video starts simultaneously with its audio track; (b) each word in a subtitle is highlighted when it is spoken in a audio commentary (synchronization between parts of media items); (c) background music is required to start with the beginning of a sequence of slides and finish at the end of the sequence.

A commonly cited ([4], [6], [13]) categorisation of temporal constraints, put forward in [2], is given in Fig. 9. These allow all possible combinations of temporal relations between two items to be expressed. More complex relations can be built up out of this set. Using the three examples above we can illustrate a number of cases. (a) the video *starts with* the audio; (b) the highlighting of the word in the subtitle *equals* the duration of the spoken word; (c) the music *equals* the sequence of slides *meeting* each other.

3.3.3. Tolerance and precision
The above temporal layout issues—such as duration of media items, synchronization constraints, interaction constraints—provide a specification of how a multimedia presentation should be played given sufficient computing resources. This is unlikely to be the case, so that variations in tolerance need to be given so trade-offs can be made. These trade-offs can be given in the form “desired timing relation, maximum allowed deviation from relation”. For example, no information content is lost if a subtitle for a video appears a second before or after its scheduled time. On the other hand, for lip synchronization only a very small deviation is acceptable. Such tolerances can be specified in percentage or absolute terms.

When defining temporal relations, with or without tolerance factors, the precision of specification can also vary. For example a delay might be specified as an absolute value such as 3 seconds or as a relative value such as “shortly after”.

For the case of spatial relations, these too could have associated tolerance and precision factors. The question is whether or not these are useful, since spatial relations are not constrained by processing power (as is the case for temporal relations), but by pixel resolution. Tolerance and precision measures may be more useful when scaling items, or when creating layouts automatically where trade-offs have to be made in how closely coupled items need to be.



Apart from the first *equals* relation, each relation has an inverse.

Figure 9. The “13” time relations.

3.3.4. Scaling

When incorporating images or video into a presentation the original size of the item may not be appropriate and a scaling operation is required, Fig. 8(a). This may be a statically defined size, such as “400 by 200 pixels”, or may be relative, such as “increase to 200%”. This deformation may take place as a function of time or be an attribute of the object. It may also be in relation to other objects, for example, increase font size until heading fits above image. This last case has not, to the authors’ knowledge, been implemented in an authoring system.

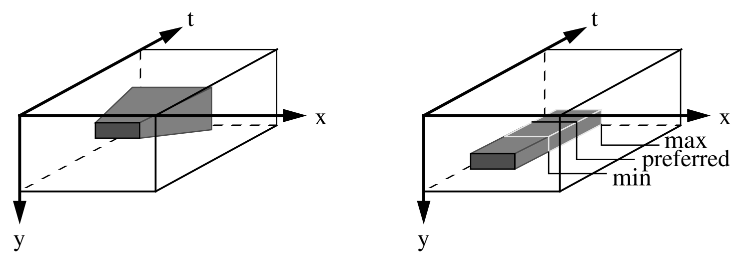
In order to satisfy temporal constraints that involve deriving the durations of objects, or satisfying constraints such as those given in Fig. 9, individual media items need to be scaled in the temporal dimension. These can take the form of specifying a preferred duration, and allowing this to be deviated from by some amount (as illustrated in Fig. 8(b)). Another form of scaling, called temporal glue, [16] and [5], allows variable length delays to be inserted into a document, so that when a media item’s duration is changed, the other constraints remain satisfied. When applying scaling factors to media items, the acceptable tolerance needs to be taken into account by the playing software.

Temporal scaling can be specified explicitly in the MET^{++} system, [1], for single and composite objects, and in the CMIFed system, [20], it is derived from the document structure. Spatial scaling is implemented in CMIFed for displaying different sized images using the same channel.

3.4. Interaction

The normal mode of operation with a multimedia presentation is that the timing relations within the presentation are calculated and the media items are displayed following the specifications. There are, however, a number of modes of interaction that can take place while the presentation is playing. These may be specified within the document, and may need the user to take an action.

A basic form of interaction is altering the speed, or direction, of the presentation as it plays. It may be played slower, faster, forwards, backwards, paused, continued, or stopped. This may be controlled by tools provided for the user, or specified as part of the document. Another common form of interaction is the hypermedia link. This specifies what happens to the currently playing presentation when the user selects to go to a different part of the presentation. This was discussed in more detail in section 3.2.3.



(a) Spatial scaling: media item can increase or decrease in size.
(b) Temporal scaling: media item has minimum, preferred and maximum duration.

Figure 10. Scaling

A more general, and consequently more complex, form of interaction can take place through conditions. These may be specified via the link structure for user interaction, or among the items themselves. For example (taken from [13]), in a presentation where the duration of the items is not specified beforehand, a conditional action might be the following. Two media items are playing and when one of the items finishes playing the other one stops, and a then third item is played. Bordegoni [4] gives a further categorisation of conditions as being deterministic or non-deterministic, and simple or compound.

The most general form of interaction can be specified with a full-blown programming language—often provided as a scripting language within a multimedia authoring system. This allows the creation of flexible and elaborate interactions, but with the consequence that they cannot be captured as part of a pre-defined document model.

3.5. Transportability

One of the goals for providing a document model for hypermedia is so that presentations can be created only once and played back on a number of different platforms, with little, or preferably no, further human intervention. This necessitates the capturing of sufficient information within the presentation description to guide the play-back process. As well as the need for a conceptual model, some way of describing the model is needed. This may be a proprietary data format, or an accessible standard, such as HyTime [11], [29] (based on SGML [33]), or MHEG [27], [28]. The scope of these standards varies. For example HyTime provides a language for expressing a hypermedia model but does not go any way towards building a system that could interpret a document description. MHEG, on the other hand, provides a set of tools for transporting packages of multimedia information.

Other work relevant to transportability is the notion of high-level resource descriptions, called channels in CMIFed [32]. When a document is played back on a particular platform, only the presentation attributes of the channel need be adapted to that platform and all the items played via the channel will conform to the attributes. Channels also allow alternative media items to be incorporated in a document description, so that when the document is played back on a particular platform one of these is chosen. For example, high quality sound takes up unwanted network bandwidth and if it is to be played back on low-cost equipment a lower quality representation could be transmitted. While this example demands the existence of numerous data representations, this process may be carried out at run time [8]. If the user's hardware supports no sound at all then a text channel might be an acceptable (although undesirable) substitution.

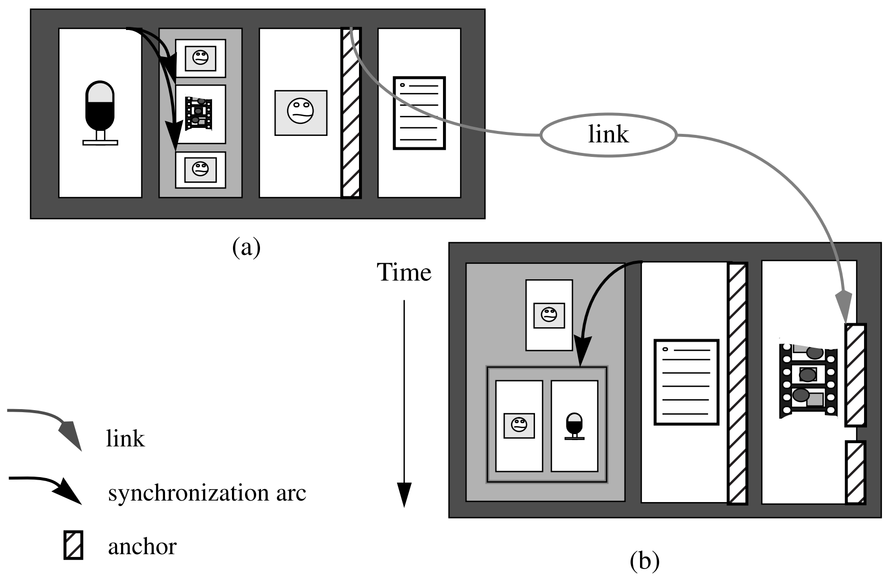
3.6. Retrieval of information

As more and more information becomes available on-line we need to find a way of finding relevant information. Classical databases are one approach, where the classification of the contained information is given beforehand by means of a schema. With widely distributed sources of multimedia information this approach is insufficient, and some other way is needed of labelling information so that it can be found again.

Multimedia information may be classified by having a library catalogue pointing at relevant items, but also by labelling items with entries from the library catalogue (similar to current book and article classifications). A hypermedia structure should not define what this labelling information should be, but should provide hooks for attaching classification information. Labelling can be carried out at the media item or even the anchor level. Burrill et al. [10], for example, define video anchors corresponding to real-life objects and these are annotated with a description of the objects they represent. It is not clear whether these annotations are part of an overall domain description. Work done by Davis [12], on the other hand, uses a rich domain description for annotating video clips for retrieval. These descriptions need not increase the storage problems significantly, since although for text nodes the amount of classification information may be larger than the text data, for video nodes this becomes an insignificant percentage.

4. The Amsterdam Hypermedia Model

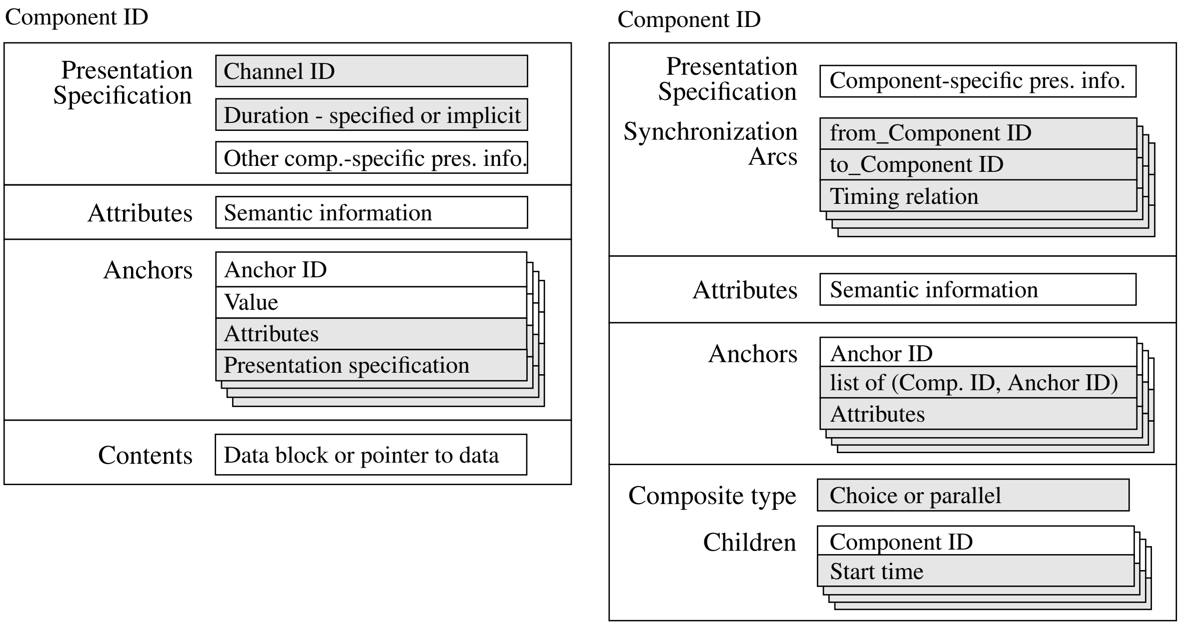
The Amsterdam Hypermedia Model (AHM), [21][2], is based on models for both hypertext and multimedia. The Dexter hypertext reference model, [15], developed by a group of hypertext system designers, describes the required structural relations. Our previous work on CMIF, [9], describes a model for specifying timing relations between collections of static and dynamic media composed to form multimedia presentations. A diagrammatic impression of the AHM is given in Fig. 11 which we will refer to in the following subsections describing the main elements of the model, discussing them in terms of the issues in the previous section.



The main components of the model are: *atomic component* (white box), *composite component* (shaded box), *link, anchor,* and *synchronization arc.* See text for explanations.

Figure 11. Amsterdam Hypermedia Model

2. Note that the model described here has been updated since that published in [21].



(a) Atomic component (b) Composite component

Figure 12. AHM atomic and composite components.

4.1. Structural information

The structure items of the model are composition, anchors and links, illustrated in Fig. 11.

- *Components* can be atomic components, link components or composite components. An atomic component, shown in Fig. 11 and specified in detail in Fig. 12(a), describes information relevant to a single media item. A link component, shown in Fig. 11 and Fig. 13, describes the relations between two components. A composite component, Fig. 11 and Fig. 12(b), is an object representing a collection of any other components.
- *Atomic component*, Fig. 12(a), includes the data needed for displaying the media item; information on how the media item is displayed (a default may come from a structure such as a channel with overrides per individual component); a duration calculated from the data type, assigned by an author or deduced from the structure, discussed in section 3.1; semantic attributes enabling retrieval of a media item, discussed in section 3.6; and a list of anchors.
- *Composition*, section 3.2.1 and Fig. 12(b), specifies the components belonging to the composite object and the timing relations among them[3]. For example, whether the grouping is time-dependent (parallel) or time-independent (choice), as dis-

3. The Dexter model also includes (a reference to) data in a composite component. This makes the presentation specifications (and other attributes) apply ambiguously to either the data in the current component or all the descendants of the component.

cussed in section 3.2.1. For a time-dependent composition, obligatory timing relations are specified through a child's start time (relative to the parent component, or to a sibling component), and optional relations among any two descendant components using synchronization arcs. The range of expression of synchronization arcs is discussed in section 4.2. Other presentation information can be used to apply a transformation to all the components in the group, without having to alter the presentation specifications of the constituent components. This might, for example, be an increase in the rate of play. Attributes allow the attachment of semantic information to components for future retrieval.

- *Anchors*, section 3.2.2 and Fig. 11, are a means of indexing into the data of a media item and are used for attaching links to components. The media-dependent data specification (the value) is specified as part of an atomic component, Fig. 12(a), and referred to from a composite anchor, Fig. 12(b), and from a link component, Fig. 13. Attributes can be attached to individual anchors for information retrieval, as discussed in section 3.6. A presentation specification per anchor allows, for example, different colours to be used for highlighting different anchor types, or for blinking an anchor of an author-specified "preferred link". Semantic information is also useful for a composite anchor, since it might represent a higher level concept. For example, individual anchors labelled as "cat", "dog" form a composite anchor labelled as "pet".
- *Links*, section 3.2.3 and Fig. 11, enable an end-user to jump to information deemed by the author to be relevant to the current presentation. The link itself is a component, Fig. 13, with its own presentation specification, attributes and anchors. The presentation specification can be used for recording transitions, section 3.2.3, where this may simply be *continue*, *pause* or *replace* for the source context, or (in the replace case using the same screen area) a more complex specification of "dissolve to new context in 10 seconds". The attributes allow link typing to be stored, such as "is a generalisation of", "isa", "is an example of". Since a link is also a component it too can have anchors referring to parts of the component.
  The link component refers to a number of anchors in other components via a list of specifiers. Each specifier refers to an anchor in a component, states possible directions for following the link, and a context that is affected on leaving or arriving at the anchor. A link most commonly has two specifiers, with a FROM and a TO direction, allowing a traversal from the FROM context to the TO context. A link may have multiple specifiers, each with its own context, so that when the link is actioned a number of currently running presentations can be stopped and a new selection begun. A presentation specification in a specifier refers to the presentation of an anchor when the link is followed, for example the part of the media item referred to by the source anchor turns grey (to indicate that the interaction has been started) and that in the destination context blinks, to highlight specific information within the new context.

4.2. Temporal layout information

Timing relations in the AHM can be defined between atomic components, composite components or between an atomic component and a composite component. For example, in Fig. 11(a) both delays are specified between two atomic components where a

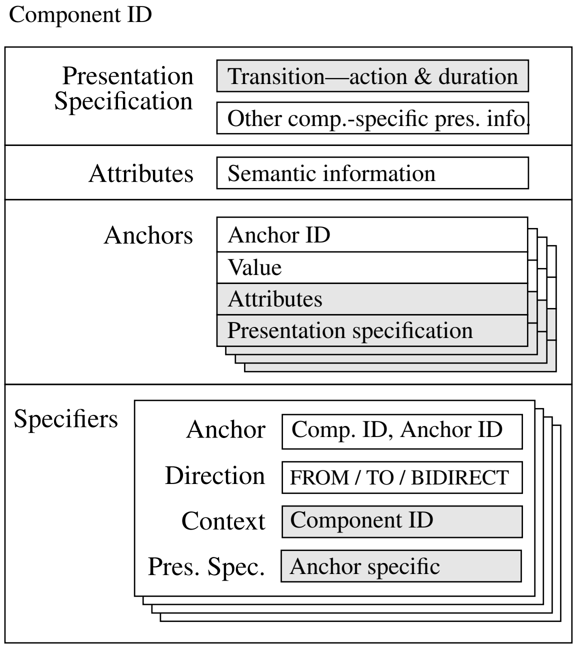


Figure 13. AHM link component.

video and image are timed to the spoken commentary. In Fig. 11(b) the delay is specified from a text item to the beginning of a composite component containing an image and spoken commentary. This timing information is stored either as the start time of a child component, or as a list of timing relations among the descendants of a composite component. This allows the timing of a presentation to be stored within the document structure itself and not as some unrelated data structure (such as a separate timeline). Note that the model specifies no boundaries on these timing relations—it is possible to specify a negative delay, so that an item can start before the whole group starts. These timing relations are specified in the model as *synchronization arcs*, Fig. 14. These can be used to give exact timing relations, but can also be used to specify tolerance and precision properties as discussed in section 3.3.2. The end of a synchronization arc may be a component, but may also refer to a (single) anchor within a component, allowing constraints to be specified within media items.

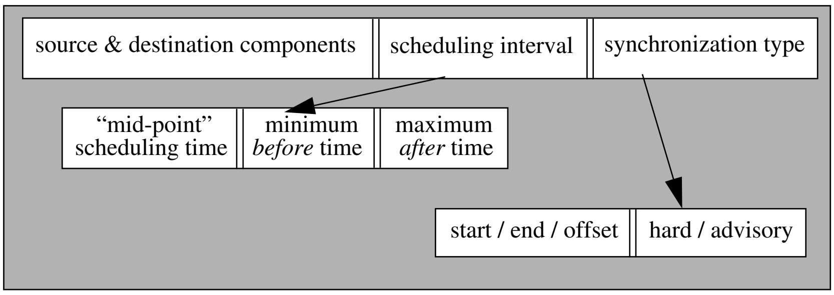
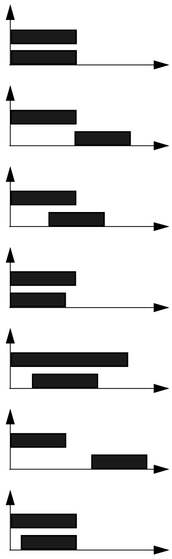


Figure 14. AHM synchronization arc.



A equals B	from beginning A, to beginning B, delay=0 *and* from end A, to end B, delay=0
A meets B	from end A, to beginning B, delay=0
A overlaps B	from beginning A, to beginning B, delay=t_1 *and* from end A, to end B, delay=t_2
A starts with B	from beginning A, to beginning B, delay=0
A during B	from beginning A, to beginning B, delay=t_1 *and* from end B, to end A, delay=t_2
A before B	from end A, to beginning B, delay=t
A ends with B	from end A, to end B, delay=0

Figure 15. The "13" time relations expressed as synchronization arcs (t, t_1, t_2 >0).

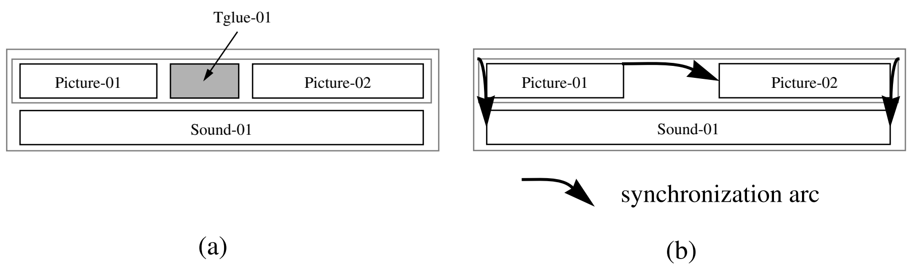
Synchronization arcs can be used to express all 13 timing relations from Fig. 9. These are illustrated in Fig. 15. Temporal glue, [16], can also be expressed in the form of sync arcs, illustrated in Fig. 16.

4.3. Spatial layout information

Layout information, such as that discussed in section 3.3.1, can be attached to each atomic component, but this can also be done through the use of *channels,* section 3.3.1, allowing high-level presentation specifications to be defined. The model does not support explicitly the inclusion of spatial relations (as an analogue to synchronization arcs), but these could be included as a specialization of the presentation specification.

4.4. Summary

The Amsterdam Hypermedia Model has been designed to capture the elements of structure, timing, layout and interaction that are needed to specify an interactive, time-



(a) Temporal composition with temporal glue as introduced object.
(b) Equivalent expressed as a number of synchronization arcs.

Figure 16. Temporal glue as synchronization arcs.

based multimedia presentation. The model does not prescribe the form these relationships should take, but implies that they should exist, and proposes how to combine them with each other. Further work has been carried out on providing a formal description of the model, [31].

Once a presentation has been described using our model, its description can be expressed in different intermediate forms suitable for multiple presentation environments. While we do not prescribe a language for specifying a presentation conforming to our model, it could be expressed in a system-independent language such as the HyTime (Hypermedia/Time-based Document Structuring Language) international standard [25], [29][4], based on SGML (Standard Generalized Mark-up Language) [33]. Initial work on converting the CMIF document format to HyTime is reported in [17]. Similarly, transmission of (groups of) multimedia objects conforming to the model could be carried out using a standard such as MHEG [28].

5. Conclusions and Future Directions

A hypermedia presentation can be viewed as a collection of media items related by structural and presentation constraints. In order to capture the essence of a presentation for playback by systems other than that it was created for, we require some sort of model of the information structure. In this chapter, we provide not only a particular document model, but have discussed the different components of the model and how these can be used for different purposes. For example, the model can be used to compare the expressive power of hypermedia authoring systems, and for transporting presentations from one platform to another. Components of the model discussed are the media items and their properties; document structure including composition, anchors and links; spatial and temporal presentation issues, tolerance and precision, and scaling; reader interaction with the presentation; transportability of complete documents; and information retrieval.

An objective of the discussion on the model components is to give examples of their possible uses, for instance possible interpretations of an anchor's presentation specification. Another objective of the discussion was to understand the limits of our model and how it would relate to other possible models. The need for specifying a particular model comes from the desire to build a concrete authoring system (in our case CMIFed).

The Amsterdam hypermedia model combines the structural features of the Dexter model with the synchronization features of CMIF. The model is sufficiently powerful to describe the documents created by current multimedia and hypermedia authoring systems. Newer features not yet implemented in our own or other systems, such as spatial constraints between components, are not excluded from the AHM, but do not appear in the model explicitly.

The model enables document transportability by allowing the translation of the model to a particular language, for example HyTime or MHEG. An initial effort has already been carried out for the HyTime case. Attributes within the model allow for

---

4. Note that the details in the published standard, [25], have superceded those described in the journal article, [29].

semantic labelling of media items, and larger structures, which can be used for multimedia information retrieval.

The limits of the AHM are reached in the case of interactivity, where links are the only means of specifying interactions. More complex interactions, such as conditionals, would require an extension to the model. Such interactions may, however, be regarded as outwith the scope of a document model and more in the realms of generalised human computer interaction.

One of the directions in which we wish to continue this work concentrates on the semantic labelling not only of components but also of parts of components (via the anchor structures). This would firstly allow information retrieval for non-textual items. Building on such techniques, we would be able to go beyond pure searching and start combining the results of searches to produce new multimedia presentations—authoring using topic content rather than specifying explicitly every spatial and temporal constraint in the presentation. We have already carried out initial design work in this direction [19], [37].

Acknowledgements

Jacco van Ossenbruggen has stimulated a detailed examination of the Amsterdam Hypermedia Model and provided many useful comments on this chapter. Guido van Rossum, Jack Jansen and Sjoerd Mullender designed and implemented the authoring environment CMIFed.

References

1 P. Ackermann (1994). Direct Manipulation of Temporal Structures in a Multimedia Application Framework. In Proceedings: *Multimedia '94*, San Francisco, CA, Oct, 51 - 58.

2 J.F. Allen (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26 (11), Nov, 832- 843

3 B. Arons (1991). Hyperspeech: Navigating in Speech-Only Hypermedia. In Proceedings: *ACM Hypertext '91*, San Antonio, TX, Dec 15-18, 133 - 146.

4 M. Bordegoni (1992). Multimedia in Views. CWI Report CS-R9263, December 1992. `<http://www.cwi.nl/ftp/CWIreports/AA/CS-R9263.ps.Z>`

5 M.C. Buchanan and P.T. Zellweger (1993). Automatically Generating Consistent Schedules for Multimedia Documents. *Multimedia Systems*. 1:55-67.

6 M.C. Buchanan and P.T. Zellweger (1993). Automatic temporal layout mechanisms. In Proceedings: *ACM Multimedia '93*, Anaheim CA, Aug, 341-350.

7 J.F. Koegel Buford (1994). Multimedia File Systems and Information Models. In J.F. Koegel Buford (ed.). Multimedia Systems. Addison-Wesley, New York, New York. ISBN 0-201-53258-1, 265 - 283.

8 D.C.A. Bulterman and D.T. Winter (1993). A Distributed Approach to Retrieving JPEG Pictures in Portable Hypermedia Documents. In Proceedings of IEEE International Symposium on Multimedia Technologies and Future Applications, Southampton, UK, April, pp107 - 117.

9 D.C.A. Bulterman, Guido van Rossum and Robert van Liere (1991). A Structure for Transportable, Dynamic Multimedia Documents. In Proceedings of the Summer *USENIX* Conference, Nashville, Tennesse, 137-155.

10 V. A. Burrill, T. Kirste and J.M. Weiss (1994. Time-varying sensitive regions in dynamic multimedia objects: a pragmatic approach to content-based retrieval from video. *Information and Software Technology* Journal Special Issue on Multimedia 36(4), Butterworth-Heinemann, April, 213 - 224.

11 L. Carr, D.W. Barron, H.C. Davis and W. Hall (1994). *Electronic Publishing*. 7(3), 163-178.

12 M. Davis (1993). Media Streams: An Iconic Language for Video Annotation. Telektronikk 5.93: Cyberspace Volume 89 (4) (1993) 49 - 71, Norwegian Telecom Research, ISSN 0085-7130 <http://www.nta.no/telektronikk/5.93.dir/Davis_M.html>.

13 R. Erfle (1993). Specification of Temporal Constraints in Multimedia Documents using HyTime. *Electronic Publishing*. 6(4), 397-411.

14 S. Eun, E.S. No, H.C. Kim, H. Yoon and S.R. Maeng (1994). Eventor: An Authoring System for Interactive Multimedia Applications. *Multimedia Systems* 2: 129 - 140.

15 Frank Halasz and Mayer Schwartz (1994). The Dexter Hypertext Reference Model. *Communications of the ACM*, 37 (2), Feb, 30 - 39. Also NIST Hypertext Standardization Workshop, Gaithersburg, MD, January 16-18 1990.

16 R. Hamakawa and J. Rekimoto (1994). Object composition and playback models for handling multimedia data. *Multimedia Systems* 2: 26 - 35.

17 L. Hardman, J. van Ossenbruggen and D.C.A. Bulterman (1995). A HyTime Compliant Interchange Format for CMIF Hypermedia Documents. In press. `<http://www.cwi.nl/ftp/mmpapers/CMIF-HyTime.ps.gz>`.

18 L. Hardman and D.C.A. Bulterman (1995). Authoring Support for Durable Interactive Multimedia Presentations. Eurographics ’95 State of The Art Report, Maastricht, The Netherlands, 28 Aug - 1 Sep, `<http://www.cwi.nl/ftp/mmpapers/eg95.ps.gz>`.

19 L. Hardman and D.C.A. Bulterman (1995). Towards the Generation of Hypermedia Structure. First International Workshop on Intelligence and Multimodality in Multimedia Interfaces, Edinburgh, UK, July 1995.

20 L. Hardman, G. van Rossum and A. van Bolhuis (1995). An Interactive Multimedia Management Game. *Intelligent Systems,* 5(2-4), 139 - 150.

21 L. Hardman, D.C.A. Bulterman and G. van Rossum (1994). The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. *Communications of the ACM*, 37 (2), Feb, 50 - 62.

22 L. Hardman, D.C.A. Bulterman, and G. van Rossum (1993). Links in hypermedia: The Requirement for context. In Proceedings: *ACM Hypertext '93*, Seattle WA, Nov, 183 - 191.

23 L. Hardman, G. van Rossum, and D.C.A. Bulterman (1993). Structured multimedia authoring. In Proceedings: *ACM Multimedia '93*, Anaheim CA, Aug, 283 - 289.

24 R. Hjelsvold and R. Midtstraum (1994). Modelling and Querying Video Data. In Proceedings of the 20th VLDB, Santiago, Chile.

25 HyTime. Hypermedia/Time-based structuring language. ISO/IEC 10744:1992.

26 T.C. Little (1994). Time-based Media Representation and Delivery. In J.F. Koegel Buford (ed.). Multimedia Systems. Addison-Wesley, New York, New York. ISBN 0-201-53258-1, 175 - 200.

27 T. Meyer-Boudnik and W. Effelsberg (1995). MHEG Explained. *IEEE MultiMedia*, Spring 1995, 26 - 38.

28 MHEG. Information Technology Coded Representation of Multimedia and Hypermedia Information Objects (MHEG) Part 1: Base Notation (ASN.1). Oct 15 1994. ISO/IEC CD 13552-1:1993.

29 S.R. Newcomb, N.A. Kipp and V.T. Newomb (1991). 'HyTime' the hypermedia/time-based document structuring language. *Communications of the ACM*, 34(11), Nov, 67 - 83.

30 R. Ogawa, H. Harada and A. Kaneko (1990). Scenario-based hypermedia: A model and a system. In Proceedings: *ECHT '90* (First European Conference on Hypertext), Nov, INRIA France, 38 - 51.

31 J. van Ossenbruggen, L Hardman and A. Eliëns (1995). A Formalization of the Amsterdam Hypermedia Model. In press. `<http://www.cs.vu.nl/~dejavu/ahm-oz/index.html>`.

32 G. van Rossum, J. Jansen, K S. Mullender and D. C. A. Bulterman (1993). CMIFed: a presentation environment for portable hypermedia documents. In Proceedings: *ACM Multimedia '93*, Anaheim CA, Aug, 183 - 188.

33 SGML. Standard Generalized Markup Language. ISO 8879:1986. Amendment 1: ISO 8879:1992/AM1:1992.

34 A. Siochi, E.A. Fox, D. Hix, E.E. Schwartz, A. Narasimhan, and W. Wake (1991). The Integrator: A Prototype for Flexible Development of Interactive Digital Multimedia Applications. *Interactive Multimedia* 2(3), 5 - 26.

35 S.W. Smoliar and H. Zhang (1994). Content-Based Video Indexing and Retrieval. IEEE *Multimedia*, Summer, 62 - 72.

36 N. West (1993). Multimedia Masters: A guide to the pros and cons of seven powerful authoring programs. *MacWorld*, Mar, 114 - 117.

37 M. Worring, C. van den Berg and L. Hardman (1996). System Design for Structured Hypermedia Generation. Visual Information Systems '96, Melbourne, Feb, 254 - 261.