2 Requirements for a Hypermedia Document Model

Using an example of a typical hypermedia presentation as a base, we discuss the features of the presentation and the corresponding requirements for a document model that would be able to specify the presentation. Features discussed are: media items forming the basis of the presentation, composition, temporal and spatial layout, and style information. We also discuss activation state and information retrieval and their consequences for a document model. We describe existing models for hypertext and multimedia and conclude that these are insufficient as a model for hypermedia. We state the list of requirements for a comprehensive hypermedia document model.
This chapter is based on work presented in [HaBu97], [HaBR94], [HaBR93b], and [HaBR93a].

2.1 Introduction

In this chapter we first specify the requirements for a hypermedia document model and then evaluate existing hypertext and multimedia models in terms of these requirements. We establish that existing candidate hypermedia models do not meet the requirements developed in this chapter. We conclude that a new model is necessary for describing hypermedia documents.

Our requirements for a hypermedia document model concern the following features of a hypermedia presentation:

- information about an individual media item,
- specification of parts of a media item,
- additional information associated with an instance of a media item when included in a presentation,
- composition of instances,
- specification of relationships among instances and compositions,
- temporal and spatial layout for instances and compositions,
- styles applicable to document elements,
- semantic information associated with instances and compositions and
- information for runtime presentation and control.

In order to clarify our discussion of the requirements for a document model we classify the requirements according to the layers of a hypermedia document processing system: characteristics of media items used in the presentation; specification of structural relationships among elements of the document; and runtime characteristics of the presentation. The first two layers have relationships between the media items and the document elements, and the second two layers have relationships between the elements and their final presentation. These layers are taken from the Dexter model [HaSc94], which, while developed for a model for hypertext, are sufficiently broad to be applicable to hypermedia document processing. Dexter terms these three layers: within-component, storage, and runtime, Fig. 2.1. The *within-component layer* stores the details of the content and internal structure of the different media items used in a presentation; the *storage layer*[1] describes the document structure; and the *runtime layer* is where user interaction is handled. The Dexter term given to the interface between the within-component layer and the storage layer is *anchoring*. The Dexter term given to the interface between the storage layer and the runtime layer is *presentation specifications*. We describe the Dexter model in more detail in Section 2.3.1.

A number of the requirements we consider in the course of this chapter are similar to those implicitly satisfied by the Dexter model or by aspects of HyTime.

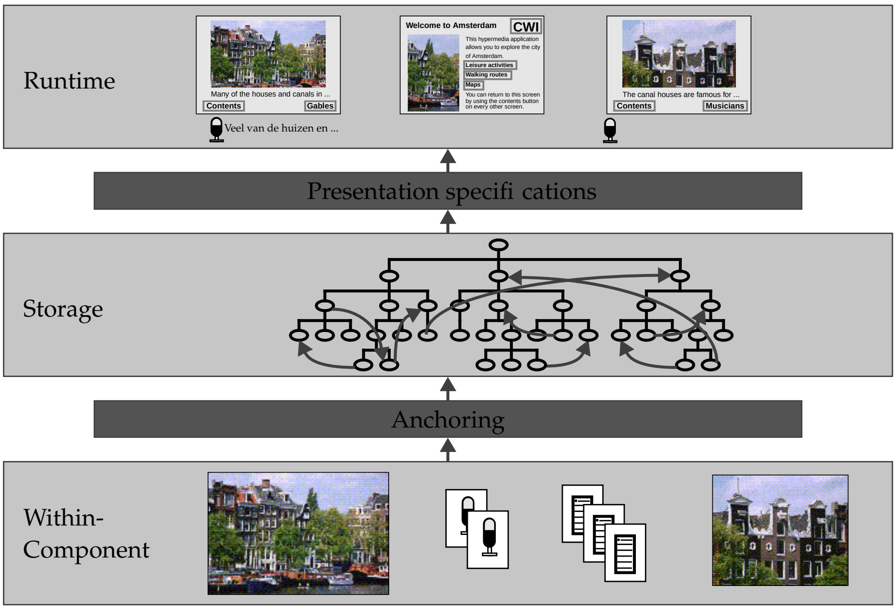


Figure 2.1. Layers of the Dexter model used for classifying requirements

1. The Dexter term storage layer is slightly misleading, since information for all of the layers needs to be stored.

The requirements tend to be expressed in Dexter terms, and so we refer explicitly to HyTime constructs where appropriate. These are included as footnotes.

We motivate our requirements for a hypermedia document model using a simple example of a hypermedia presentation. This provides an intuitive introduction to the requirements. The example presentation incorporates synchronized discrete and continuous media and allows the reader to make selections within the information. We deliberately choose a small example to provide a lower bound on the requirements necessary for describing a hypermedia presentation. Whether our requirements are suffi cient for describing presentations created by a broad range of systems is an empirical question. We postpone this question until Chapter 3, where we show that our proposed document model is able to describe the presentations created by a range of existing systems. In order to unify the various discussions in the following sub-sections, we refer back to this example throughout this chapter.

The presentation shown in Fig. 2.2 illustrates three fragments from a tour of the city of Amsterdam. The top fragment, (a), is analogous to a table of contents and provides the reader with a description of the tour and a number of options from which to select. This continues to be displayed until the reader selects another scene to be displayed. One of these is a description of a walking route through the city, highlighting a number of places of interest found on the tour. Two places of interest are shown in Fig. 2.2(b) and (c).

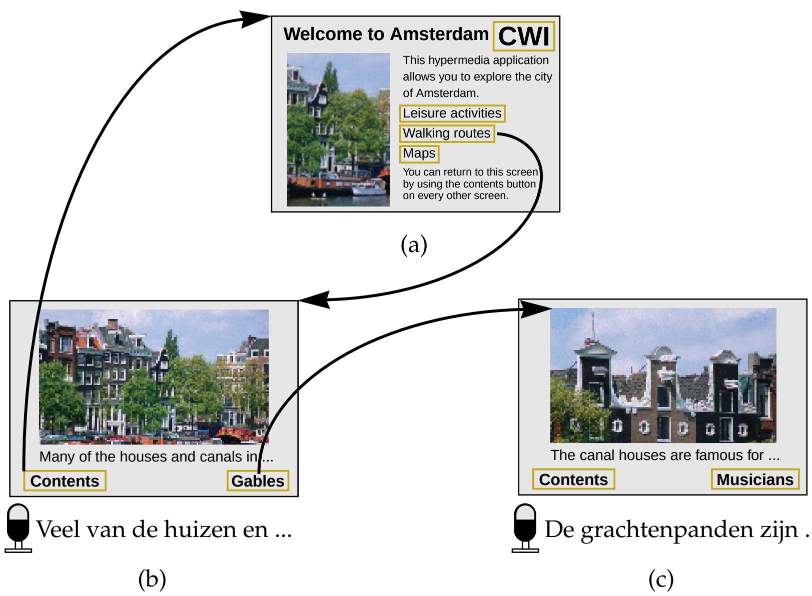


Figure 2.2. An example hypermedia presentation

For each scene to be displayed, the player requires:

- the data for each of the media items,
- the starting time and duration of each instance of a media item in the presentation and
- its extent and position on the screen.

For example, in Fig. 2.2(a) a specification for the position of the heading text is required, as well as its typeface and size. Thus the requirement for the document model is that for each media item there needs to be information about its position, the start time and duration of its display, plus style information.

Further specifications are needed for supporting navigation among presentations. For example, in Fig. 2.2(a) the initial opening scene is playing and at some unspecified point in time the reader selects to go to the walking route in (b). At this point the opening scene fades out from the screen and the first section of the walking route fades in. The action of making the selection requires information for specifying the following:

- where the reader is able to make a selection, e.g., the three boxed phrases in (a);
- where each selection leads, e.g., "Walking routes" in (a) leads to the scene in (b), and "Gables" in (b) leads to (c); and
- how the transition should be made from the scene that was playing to the newly selected scene, e.g., when going from (a) to (b) nothing remains of the presentation in (a), whereas when going from (b) to (c) the "Contents" text is common. The scene in (a) fades into the scene in (b), whereas the scene in (b) does a "wipe left" to the scene in (c).

The requirement for the document model is that for each selection information is required for specifying a part of a media item, for associating a destination with the point of selection, for specifying the scope of the presentation affected on following the link and for describing the special effect associated with making the selection.

This simple example illustrates a number of features of a hypermedia presentation that need to be specified as part of a hypermedia document model. In the following section we go into greater detail for each of these.

This chapter is structured as follows. Section 2.2 states the requirements for a hypermedia document model and discusses each of these in detail. Section 2.3 describes existing models for hypertext and multimedia, compares the requirements with the existing models and lists the limitations of the models as models for hypermedia. Section 2.4 concludes that we need a new model for hypermedia. The specification of a hypermedia model is given in the following chapter.

Dexter layers	Model features	Section
Within-component layer	Media items	2.2.1 Within-Component layer: Media items
Anchoring	Reference to part of media item	2.2.2 Anchoring
Storage layer	Properties associated with instance of media item	2.2.3.1 Instance of media item
	Composition	2.2.3.2 Composition of instances, 2.2.3.3 Composition of anchors
	Linking	2.2.3.4 Linking
	Semantic attributes	2.2.3.5 Semantic attributes
Presentation Specifi cations	Temporal layout	2.2.4.1 Temporal layout
	Spatial layout	2.2.4.2 Spatial layout
	Styles: media item, anchor, transition	2.2.4.3 Styles
	Initial activation state	2.2.4.4 Activation state
Runtime layer	Temporal flow	2.2.5.1 Temporal control
	Spatial layout	2.2.5.2 Spatial control
	Navigation (activation changes)	2.2.5.3 Navigation control

TABLE 2.1. Document model features for hypermedia

2.2 Requirements for a hypermedia document model

We use the example in the previous section as a starting point for specifying the requirements for a hypermedia presentation. We structure the discussion using the Dexter layers and go through the layers in a bottom-up manner. We fi rst describe characteristics of the media items, and then go through the problems of including and combining these into a presentation. Table 2.1 gives a summary of the document features and where they are discussed in this section. Throughout this section, the emerging document model requirements are stated and summarised in tabular form. These smaller tables are collected together at the end of the section in Table 2.13 which provides a complete summary of the document model requirements.

2.2.1 Within-Component layer: Media items

A media item contains the data that is presented to the reader and as such is the basis of a presentation. We defi ne the*media item* as an amount of data that can be retrieved as one object from a store of data objects—although not necessarily a small amount. Media items can be of different media types. For example in Fig. 2.2(a), the screen shown consists of four media items—a video of a canal scene, the CWI logo, a heading and a longer text item. Although a multimedia presentation is built up from heterogeneous pieces, it is perceived by the reader as a continuous whole. The goal is to integrate different media types into the presentation while retaining media type independence of the document model.

We first discuss the nature of a number of common media types and then discuss their dimensionality.

Media types

For the purpose of combining different media types within a single presentation, we define the characteristics of the four basic media types: text, image, audio and video.

- *Text* is an ordered linear sequence of two-dimensional characters, e.g. English language text, braille, Chinese characters.
- *An image* is a static two-dimensional, visual representation, e.g. a real world image such as a drawing or a photo, or a symbolic representation such as a graph.
- *Audio* is a continuous audible medium, e.g. speech, music, or sound-effects.
- *Video* is a continuous sequence of moving images, e.g. continuous real-world images, animation or any sequence of still images that is intended to be perceived as a unity.

A number of media types, such as text and vector graphics, can of themselves be structured. Text is a special case in that the internal structure of the media item can be expressed using the components of it, e.g. HTML [Ragg97] or Postscript®[Adob90].

A media item may consist of a single medium, such as those just described, or a composite medium such as interleaved video and audio.

Temporal and spatial dimensionality

We distinguish two categories of media—continuous and non-continuous, or discrete. Continuous media have a temporal dimension intrinsic to the media item itself, e.g. audio and video. Non-continuous media have no intrinsic temporal dimension, e.g. text and images.

Fig. 2.3 shows representations of the four basic types of media items in a three-dimensional space.

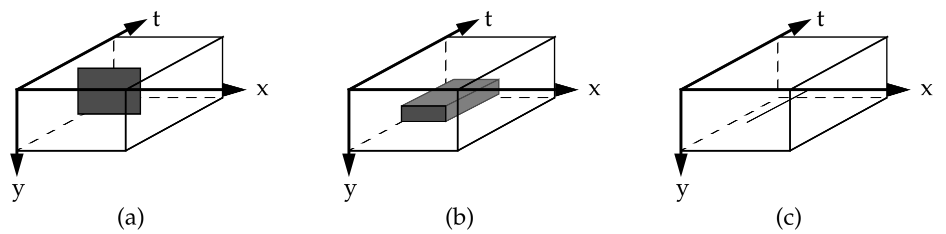
- Text requires two spatial dimensions for display, Fig. 2.3(a). The aspect ratio of the display area is relatively unimportant, since lines can be broken at various positions without altering the semantics of the message.
- Images also require two spatial dimensions, Fig. 2.3(a), but the aspect ratio is important for aesthetic reasons as well as for accurate representation of real-world objects.
- Video requires two dimensions of space plus time to be displayed, Fig. 2.3(b). Video can be regarded as a sequence of images, where each image is displayed at a particular time. The aspect ratio is thus important.
- Audio is a continuous medium and has no spatial dimensions, Fig. 2.3(c).

While we portray the presentation here as taking place in two spatial dimensions plus a time dimension, three spatial dimensions plus a time dimension is also a possibility (although representations on two dimensional paper would be difficult to interpret). For example, with virtual reality applications the media items are three dimensional objects in a three dimensional space whose position, extent and orientation can change with time. The other aspects of the model addressed in this section are similarly not restricted to three dimensions, but can be illustrated within them.

When a media item is incorporated into a multimedia presentation it is displayed on the screen for some duration or played through an audio device. Thus, in order to incorporate a media item within a multimedia presentation knowledge of its temporal and spatial dimensions is required[2].

Generated media items

Other data types that should be includable in a presentation are outputs from external programs or processes, for example the video signal from a camera pointing at an outside scene, the reading from a monitoring device in a chemical plant or power station, or computer synthesized music. For live feeds the data can be treated as having a pre-specified spatial extent but with an indefinite duration. Alternatively, data generated on-the-fly from an external program may be of known spatial extent and duration. For example, financial results generated from a market simulation are displayed as a graph in a presentation, [HaRB95]. The requirement for including live data, program code or other generated media items in a presentation is the same as for the standard data types—that the spatial and temporal dimensionality be known beforehand. While it is also useful to know the spatial and temporal extents, where the duration may be indefinite, this is not a requirement.



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

Figure 2.3. Spatio-temporal dimensions of media types

2. These coordinate spaces are similar to the concept of finite coordinate spaces (FCS) in HyTime [ISO97b].

Dexter layers	Model features	Model requirements
Within-component layer	Media items	temporal and spatial dimensions

TABLE 2.2. Model requirements for the within-component layer

2.2.2 Anchoring

The content included within a presentation does not necessarily have to be an entire media item, but might be a reference to part of it. For example the picture of the gables as used in Fig. 2.2(c) is only a part of the original picture, shown at the bottom of Fig. 2.1. This allows multiple use of the same data without the need for extra storage. The content can be given by a reference to the stored media item and a media-dependent specifi cation of a part of it. Examples for other media types are the following. In the case of text the media item may be a complete book, where only a section is required. In audio, 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).

To allow a reader to make a selection to go to another scene there needs to be information on the screen denoting where the reader can make a selection, and what the expected information at the destination will be. For example, in Fig. 2.2(a) there are three choices within the text item plus one on the CWI logo. These selections need to be specifi ed within the document.

In order to create a synchronization relationship with a point or an interval in a continuous media item a means of specifying the point within the media item is needed. For example, in Fig. 2.2(b) and (c) there is a spoken commentary. At selected points in the commentary the subtitles change. When exactly within the spoken commentary the subtitles should change needs to be described.

In all three cases, a data-dependent specifi cation of part of the media item is required. We term this the anchor value [HaSc94].

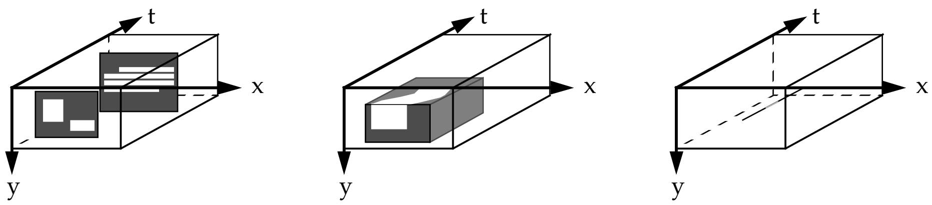
Dexter layers	Model features	Model requirements
Anchoring	Reference to part of media item	data-dependent specifi cation of part of media item

TABLE 2.3. Model requirements for anchoring

We give examples of anchor values in different media types. A graphical interpretation is given for text, image, video and audio anchors in Fig. 2.4.

- A *text anchor value* normally specifi es a sequence of characters within a text item, Fig. 2.4(a). This may be given as a character offset and length of a text string, but, to withstand possible editing of the text, this should also specify keywords such as the beginning, subject and end of the text fragment, along with extra context information about where the fragment can be found. In some cases a text query may be the most appropriate, for example stating what the text should be about rather than which letters should be part of the text.

- An *image anchor value* specifies an area in a pixel-based image, where most systems implement the area as being rectangular, Fig. 2.4(a), although there is no theoretical restriction on its shape and any contour could be defined to specify its extent—allowing objects in the image to be traced out. 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 is data format dependent and can refer to anything appropriate to the data format. Similarly an anchor could be specified in a 3D graphic as an object within the graphic, for example a house in a virtual reality landscape.
- A *video anchor value* may be specified as a sequence of frames, as is used in a number of systems [Davi93], [HjMi94]. This allows the user to select at most one link to follow to another presentation from any frame in the video. A more desirable approach is to specify the area on the screen for the extent of the frame sequence [SmZh94]. 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, [BuKW94]. This allows moving or changing objects in the video to be followed, so that clicking on an object becomes a more natural option, illustrated in Fig. 2.4(b). For a live video feed, objects within the video image could be designated as anchor values, but real-time image recognition would be needed to implement them.
- An *audio anchor value* can be specified as a time partitioning of the audio, illustrated in Fig. 2.4(c), or it might be a timed extent of one musical voice, for example a number of bars of violin solo. The problem starts when the reader tries to interact with the audio item, since with the normal mode of



(a) (b) (c)

(a) Graphics anchor, e.g. area; text anchor, e.g. text string.
(b) Video anchor, e.g. area changing with time.
(c) Audio anchor, e.g. temporal extent, or in music a temporal extent within an instrument or voice.

Figure 2.4. Anchors

interaction with hypermedia presentations (clicking an object on the screen) there is nothing tangible with which to interact—although links *to* audio items remain possible. An example of interacting with "hyperspeech" is given in [Aron91]. Here, although no anchors were used in the application, the authors suggest using a Doppler shift effect to suggest that the listener is approaching, or passing, a hyperspeech branch. The anchor value would be a portion of the audio data, and the Doppler effect the presentation style.

Anchor values can be specified for other media types, although there is less agreement on their form. For example, for a simulation program distinct states of the program could be used as anchor values.

The anchor value is specified in terms of the data format of the media item it refers to. Just as the media item has its own intrinsic spatial and temporal dimensions, the anchor value inherits these dimensions. Not only is the dimensionality the same, but the position and extent of the anchor value is defined in terms of the temporal and spatial axes defined by the media item. The consequence of this is that an anchor value cannot be scaled without also scaling the media item it refers to.

Another requirement is for the synchronization of anchors in a static medium with those in a continuous medium. Examples include a piece of text that has an accompanying spoken commentary, or a music score with its associated performance. To show the correspondence between the anchor values, an anchor value in the static medium should be able to be highlighted for the duration of the corresponding anchor value in the continuous medium.

2.2.3 Storage layer

The storage layer is where the media independent structure of the presentation is stored. While the media item is the basis of a presentation, information needs to be associated with each instance of a media item in a presentation. An *instance* is the inclusion of a media item in a presentation. An instance requires the specification of the media item along with properties associated with its inclusion. In order to create scenes, collections of instances need to be specifiable. In order to make selections among scenes there needs to be a construct to store the relationship. In this section we discuss the requirements for an instance, composition, and for making selections among scenes.

2.2.3.1 Instance of media item

When including a media item in a presentation information in addition to the data for the media item is required. This may be a specification of which part of the media item is displayed in the presentation, as discussed in the previous section on anchoring, or other types of information. For example, in Fig. 2.2(a) the position and extent of each of the media items, the font typeface and style for the text items and an appropriate background colour is needed. In Fig. 2.2(b) the subtitles change with time, so that some specification of when each subtitle should be displayed is also required.

The list of properties required for specifying an instance of a media item are:

- a reference to the storage of the data object for the media item and its data format;
- a specification of the part of the media item which is to be presented;
- duration and start time of the media item;
- extent and position of the media item.

These properties are required, since without some knowledge of their value the player is unable to display the instance at the appropriate position and time in the presentation. The list of properties that may be associated with an instance of a media item are:

- aspect ratio;
- orientation;
- Z-order, i.e. the front to back ordering of media items;
- style;
- start points for choices of destination;
- semantic information for finding media items.

The duration or extent of a media item may be intrinsic to the media type of the item. If this is the case, then the duration or extent of the item as incorporated in the presentation may require some other value. This may be in terms of a scale factor or an absolute value. Start time and position cannot be specified for a media item in isolation from the rest of the presentation but need to be specified in relation to other media items or with respect to the presentation as a whole. As a consequence, these cannot be stored as properties along with the media item, but as part of the structure of the presentation. We discuss temporal specifications further in Section 2.2.4.1.

The aspect ratio, orientation and Z-order are applicable to screen-based media. We discuss these further in Section 2.2.4.2.

The style of presentation of the media item is dependent on the media type, and includes font size for text, or a colour map for images. We discuss style in more detail in Section 2.2.4.3.

In order to incorporate selection points within a presentation, there needs to be some way of recording where these start points should be. We discussed this in Section 2.2.2 on anchoring.

While not obligatory for the display of a hypermedia presentation, some means for finding relevant parts of a presentation is useful for authors creating presentations out of pre-existing parts, or for readers looking for specific pieces of information. Attaching semantic information to a particular use of a media item allows that use to be categorised, indexed and searched upon. We discuss this in more detail in Section 2.2.3.5 on attributes.

In a model for hypermedia the encapsulation of a media item in an instance should be expressible along with a means for associating other required and optional properties of that use of the media item as part of the presentation.

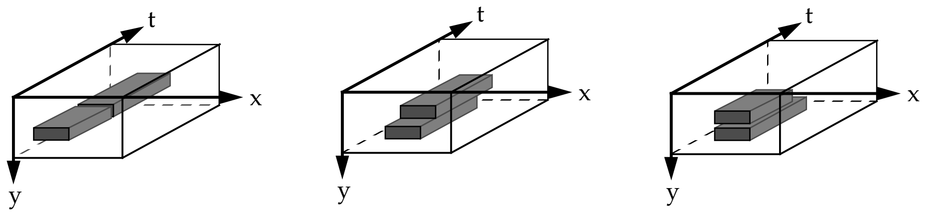
Dexter layers	Model features	Model requirements
Storage layer	Instance of media item	reference to (part of) media item, data format, duration, start time, extent, position, aspect ratio, orientation, Z-order, style (media item, anchor, transition), start points for links, semantic attributes

TABLE 2.4. Model requirements for an instance of a media item

2.2.3.2 Composition of instances

One of the distinguishing characteristics of a multimedia presentation is that it consists of a number of instances of media items combined together into an integrated presentation, for example, each of the three scenes shown in Fig. 2.2. A hypermedia model thus requires a composition mechanism for defining which instances are included in the presentation and how they are combined together, in particular their temporal and spatial relations. We name this type of composition space/time-dependent composition. *Space/time-dependent composition* specifies a number of children, along with temporal and spatial information relating the children. This type of composition allows the merging of smaller presentations into a larger presentation, Fig. 2.5. The timing of the children and the spatial aspects have to be specified in terms of the same coordinate axes[3].

Space/time-dependent composition alone is insufficient for describing a hypermedia application, since an application can consist of more than one multimedia presentation, as illustrated by the separate scenes shown in Fig. 2.2. These separate presentations need to be associated together in some way, but without merging them into a larger presentation. A second composition mechanism is thus required which allows the composition of multiple presentations. We name



(a) Sequential (b) Intermediate (c) Parallel

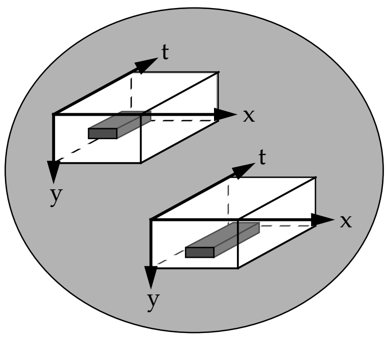
Figure 2.5. Space/time-dependent composition

3. Space/time dependent composition of instances is similar to the HyTime [ISO97b] placement of events within the same finite coordinate space.

this type of composition space/time-independent composition. *Space/time-independent composition* requires the specification of a number of children, but with no other temporal or spatial information. This type of composition allows the inclusion of separate presentations in a hypermedia document, Fig. 2.6. If any one presentation is playing then the only method of playing another is to follow a link to it. If no link exists to a presentation then it can never be played, since it is not part of the flow of some larger presentation. There is no intrinsic temporal relation among the presentations, and it cannot be predicted exactly when, or indeed if, the reader will follow a link to any one of them.

An example of space/time-independent composition is where multiple presentations can be played simultaneously. For example, when the reader selects the CWI logo in Fig. 2.2(a) a spoken commentary is given about the institute. The start time of the spoken commentary is not bound to the presentation already playing, but is conditional on the reader following the link from the logo. The time bases of the two presentations are independent within the document specification, but the presentations are played simultaneously when the reader follows the link.

Two other types of composition can also be identified: temporal composition and spatial composition. *Temporal composition* is time-dependent but space-independent; *spatial composition* is space-dependent but time independent. Two important categories of temporal composition are parallel and sequential composition ([Acke94], [HaRe94], [HaRB93]), Fig. 2.5(a) and (c). In the parallel case, the temporal relation is that the instances start together; in the sequential case that one instance starts when the previous finishes. These are sometimes treated as fundamental divisions, although they are two extremes of temporal composition. An intermediate case is that one instance starts and at some time later, but before the first finishes, the second one starts, Fig. 2.5(b). An example of using



A composite element, denoted by the shaded ellipse, contains two sub-elements aligned along independent coordinate axes.

Figure 2.6. Space/time-independent composition

temporal composition with spatial independence is where a presentation is playing and after a pre-specified time a second presentation starts to play in another window. The reader is able to move the positions of either window independently.

An example using spatial composition is where a presentation is built up of several subscenes. A number of items are able to remain on the screen (e.g. in Fig. 2.2(b) a link back to the contents screen) while the reader selects the different subscenes (e.g. in Fig. 2.2(b) a picture with spoken commentary and a text item linking to the next subscene). There is no pre-specified temporal relation between the items that remain on the screen and those playing in the subscenes. The spatial relation is, however, fixed. Another example of spatial composition is described in [Grøn94], termed a table top composite, where the spatial relations among the child elements are recorded.

In a model for hypermedia temporal, spatial and space/time-independent composition are essential, fundamental structuring mechanisms and should be expressible. Space/time-dependent composition is an alternative to separate temporal and spatial composition mechanisms. These requirements are summarised in Table 2.5.

2.2.3.3 Composition of anchors

A multimedia presentation is, by definition, assembled from a number of items of different media. These items, however, may contain illustrations of the same concept. For example, in Fig. 2.2(b) the word "houses" in the subtitle, the houses in the picture and the spoken Dutch word "huizen" within the commentary all illustrate the concept "houses". A means is required for grouping the different expressions of the concept as a single element. This would allow the creation of a single relationship from the "houses" concept rather than forcing the creation of multiple relationships with otherwise identical properties.

The grouping of anchors with similar concepts is orthogonal to the temporal/atemporal structuring of the presentation. For example, the three anchors described above happen to be displayed at the same time, but there may be other illustrations of the concept "houses" which occur at other points in the presentation. These may also form part of the single conceptual anchor.

Note that composition of anchors is similar in spirit to generic links in Microcosm [HaDH96].

The model requirement is that parts of different media items expressing a single concept should be able to be treated as a single element.

Dexter layers	Model features	Model requirements
Storage layer	Composition	temporal and spatial composition, space/time independent composition, grouping of anchors into composite anchors

TABLE 2.5. Model requirements for composition

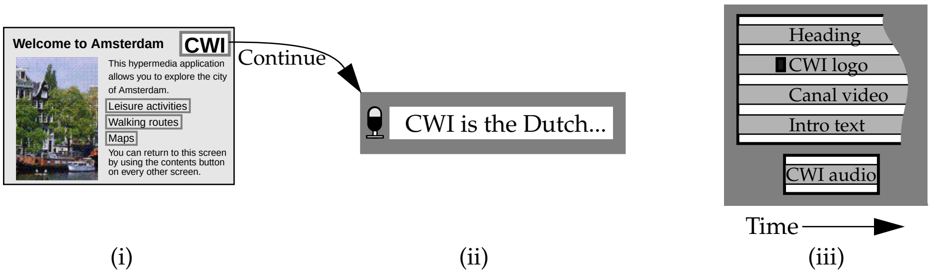
2.2.3.4 Linking

While viewing an interactive on-line presentation, a reader should be able to make choices as to which information is displayed. For example, in Fig. 2.2(a) there are four highlighted areas from which the reader can choose. When one of these is selected the destination of the link is displayed. For example, by clicking on "Walking routes" in (a), the introduction is replaced by the scene in (b). In the case of hypermedia, where multiple items are playing simultaneously, they may not all be replaced, but some subset of them. For example, by selecting "Gables" in (b), not all of the scene is replaced by the scene in (c)—the "Contents" text remains playing. In this case, the scope of the information associated with the link is a part of the original presentation. We call this scope specification the *link context* [HaBR93b], [NaNa93]. The *source context* is the information associated with the source of the link and the *destination context* that associated with the destination of the link. There may be multiple source and destination contexts, but for the sake of simplicity we discuss the situation of only one source and one destination context.

Not only does the source context need to be specified, but also what should happen to the source context when the reader follows the link. There are three options which we illustrate in Fig. 2.7. The source context can remain playing, so that the destination context is a new, independent, presentation which is started up, Fig. 2.7(a). The source context can remain on the screen but pause, (b). The source context can be cleared from the screen and the destination context played on its own, (c). Note that when following a link the source and destination contexts become part of the same virtual temporal composition when the reader makes a link selection. The information needed to specify the link transition is the same as that needed when grouping elements to become part of the same temporal composition.

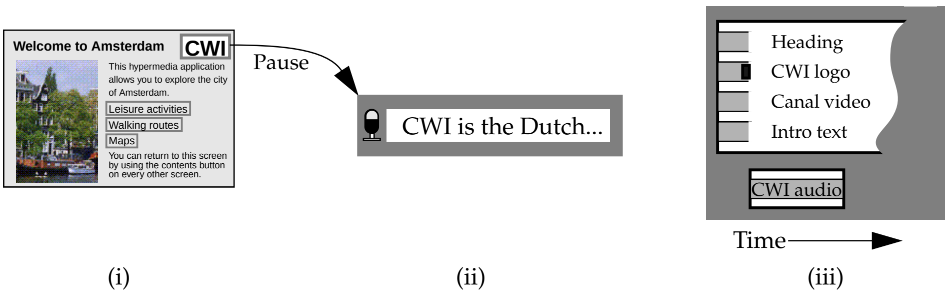
When a reader follows a link from the source to the destination context, this should not introduce a break in the flow of the presentation, but should be a smooth transition from one scene to the next. The action of going from the source to the destination context we call the *link transition*, [SFHS91]. A link transition has a duration and an associated style. The duration is the length of time it takes to play the transition. The style is a special effect that can be applied to the source context as it changes into the destination context, for example, effects such as "wipe-left", "zoom-out", "dissolve", or sound effects. The transition might even be a sequence in its own right. For example, if a user chooses to visit 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.

When the link has been followed and the reader arrives at the destination, it can be useful to highlight a particular object at the end of the link. This can be achieved by specifying a destination anchor which can be highlighted in a man-



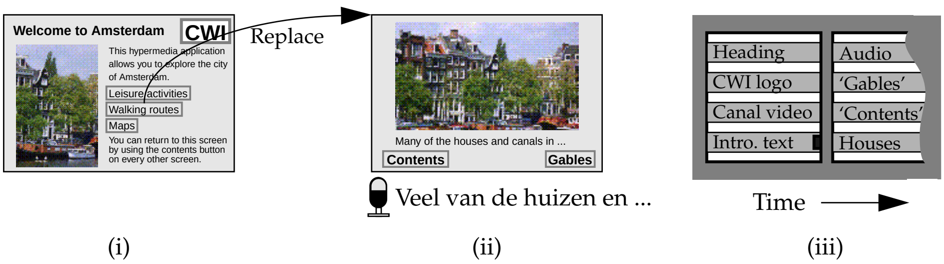
(a) Continuing the source context

While the scene in (i) is playing the reader clicks on the CWI logo. The scene continues while a spoken commentary (ii) is delivered. The same interaction is seen from a time-based point of view in (iii).



(b) Pausing the source context

While the scene in (i) is playing the reader clicks on the CWI logo. The scene pauses and a spoken commentary (ii) is delivered. The same interaction is seen from a time-based point of view in (iii).



(c) Replacing the source context

The reader clicks on the *Walking route* text and the scene is replaced by the first scene of the walking route. The same action is seen from a time-based point of view in (iii).

Legend: Playing item Reader clicks on anchor Paused item

Figure 2.7. Runtime behaviour of links in hypermedia

ner similar to, though not necessarily the same as, highlighting the source anchor. For example, when going from the walking route scene in Fig. 2.2(b) to the introductory screen in Fig. 2.2(a) the "Walking routes" anchor marker could be flashed once to indicate that is where the reader just came from.

In summary, when specifying a link in hypermedia the following is required:

- a source anchor that the reader can select in order to choose the destination;
- a specification of how much of the displayed presentation belongs to the source link context;
- a description of how the source context is transformed into the destination context, and whether it remains playing;
- a specification of the destination link context.
- A destination anchor may also be associated with the link to emphasize a particular part of the destination context.

Dexter layers	Model features	Model requirements
Storage layer	Linking	source and destination anchor, source and destination context, transition (duration and special effect) change in activation state change in playing/paused state

TABLE 2.6. Model requirements for linking

2.2.3.5 Semantic attributes

As more and more distributed sources of multimedia become available, some way is needed of labelling the information stored in these sources so that it can be found again. A further requirement is that presentations are needed for different circumstances, such as reader information requirements, display platform or available network bandwidth. Rather than creating these presentations individually, authoring effort can be spared by generating them from an underlying representation. For both information retrieval and for more automated authoring, some connection with the semantic content of the presentation has to be made.

One means of merging a multimedia presentation with a knowledge representation is to associate semantic labels, which we call *attributes*, from a knowledge representation with parts of the document structure. This is analogous to labelling a book with classifications from a library catalogue. A hypermedia document model should not define what the semantic attributes should be, but should provide hooks for attaching classification information. Alternatively semantic information may be associated with a multimedia presentation by having a knowledge structure refer to media items expressing a particular concept. This is analogous to a library catalogue listing the books it holds in each category. In both cases it is a many to many mapping, where each book or media item can be associated with multiple categories or concepts, and each category or concept can be associated with multiple books or media items.

While the library classifi cation example does not go further than the book as a unit of classifi cation, in the case of hypermedia the labelling should be carried out for single media items, for collections of media items and for parts of media items [NaNa93], in particular for larger media types such as video [BuKW94].

By labelling media items with semantic attributes, fragments of presentations can be found that correspond with a reader's information need. Having retrieved the appropriate fragments, a larger presentation can be generated from them. For example Davis [Davi93], generates sequences of video from a store of labelled video clips and Worring et al. [WBHT97], specify the design of a system for generating hypermedia presentations on the basis of semantic labelling. Alternatively, the presentation can be generated top down from a knowledge-based description, e.g. André et al. [Andr96] generate multimedia presentations on the basis of domain information.

Although a hypermedia document model should not specify the form of the attributes, it is useful to give an illustration of the way we expect attributes to be used.

- *Anchors*
  Anchors in instances correspond to the basic semantic objects which can be seen by the reader and described by the author. There will no doubt be higher level, or more abstract, concepts involved in the presentation, but these are more diffi cult to point at directly. For example, a bicycle and wheel shown in a video are labelled with the semantic attributes " bicycle" and " wheel".
- *Instance*
  For an instance it is likely that there is a collection of attributes already associated with anchors of the instance. There need be no extra attributes associated with the instance, but there may be a higher-level abstraction associated with it. For example, an instance referring to an image with two anchors labelled " funny hat" and " cake" may have the attribute " birthday party".
- *Composition*
  Similarly, for a composition it is likely that there is a collection of attributes already associated with the descendants of the composition. An attribute associated with the composition may then be a higher-level, or more abstract, term.
- *Composite anchor*
  For a composite anchor, there is a collection of attributes similar to the collection for anchors in an instance, although the same attribute may be expressed in different media. For example, the word " bicycle" in a text item has the attribute " bicycle" and a picture of a bicycle has the same attribute.

The composite anchor referring to both anchors still has the same attribute " bicycle".

- *Link*
  The attributes associated with a link are slightly different from those associated with the other objects. A link specifies a relationship among objects, and thus the attributes would be expected to reflect this. For example, a link from an anchor with the attribute " wheel" to one with " bicycle" would have the attribute " is-part-of". A link from an anchor with the attribute " mountain bike" to the same " bicycle" anchor would have the attribute " is-a-type-of".

In conclusion, a hypermedia document model should allow semantic information to be associated with anchor values, instances, compositions, composite anchors and links.

Dexter layers	Model features	Model requirements
Storage layer	Semantic attributes	associate with anchors, instances, compositions, composite anchors and links

TABLE 2.7. Model requirements for semantic attributes

2.2.4 Presentation specifications

The presentation specifications form the interface between the storage layer and the runtime layer. Presentation specifications for multimedia include temporal and spatial layout and style. Temporal layout is fundamental to multimedia and is the specification of when an instance is presented and how long it remains playing. Spatial layout is the specification of the position and extent of the instance on the screen. Style information includes the choice of colour, font, etc. for the various media items used within the presentation, the signalling of anchors and transition styles.

2.2.4.1 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. We 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. The six subsections address: defining a time axis in relation to which instances can be played, the start time, duration and temporal scaling of an instance, temporal relations among instances and the temporal relation associated with a link transition.

Specification of time axis

Including an instance within a multimedia presentation requires specifying when it should be played in relation to some implicit or explicit time axis. A common example of an explicit time axis is the timeline, Fig. 2.8(a), as used in Director [Macr97] and the Integrator [SFHS91]. A time axis can be defined implicitly when durations of instances are known and these are grouped together with known temporal relations among them. A number of systems use this method to derive a time axis: e.g. Eventor [ENKY94], Mbuild [HaRe94] and CMIFed [HaRB93].

In either case the rate of traversal along the timeline can be varied, in a similar way that music can change its tempo when being performed.

The requirement for a hypermedia model is that a time axis should exist, whether it is specified explicitly or is implicit.

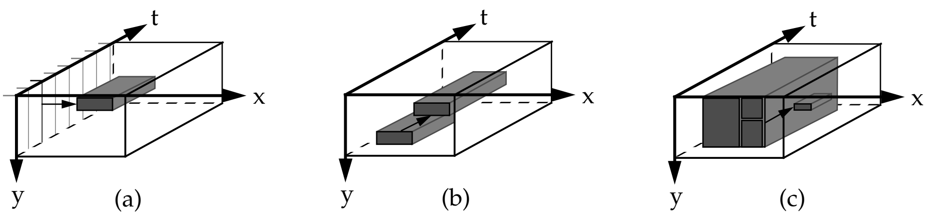
Start time of instance

The start time of an instance can be given in a number of ways. The most common method is to define an instance's start time relative to a timeline, illustrated in Fig. 2.8(a). Examples of authoring systems using this approach are the Integrator [SFHS91] and Director [Macr97]. A second method is to specify the start time with respect to another instance in the presentation, illustrated in Fig. 2.8(b). This is supported, for example, by Firefly [BuZe93a] and Eventor [ENKY94]. A third possibility is to specify it with respect to a composition, illustrated in Fig. 2.8(c) and implemented in MET^{++} [Acke94].

The requirement for a hypermedia model is that the start time for an instance is expressible. This can either be specified explicitly or can be calculated from relations with other parts of the document structure.

Duration of instance

As well as specifying when an instance should begin, its duration also needs to be known. This may be specified explicitly or implicitly or derived from relations with other instances. For example, a video has an intrinsic duration associated with its media item. An image, however, needs to be assigned a duration. This may be explicitly assigned or derived from the surrounding presentation. An example of a derived duration is where the image is displayed when a spo-



(a) Start time is specified with respect to a timeline.
(b) Start time is specified with respect to another instance.
(c) Start time is specified with respect to a space/time-dependent composite.

Figure 2.8. Ways of specifying start time

ken commentary begins and remains 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 along the time axis.

An anchor of a media item may also have an associated duration. The start or end time may be specified by a point in time, e.g. a frame number in a video, or “5 seconds after the start” of an image, or “the beginning of the word gables” in a spoken commentary. The duration can be given by an interval, e.g. a number of frames in a video, “between 3 and 5 seconds” for an image, or the time it takes to say the word gables in a spoken commentary. Alternatively, the duration may be specified using begin and end times, e.g. “from the beginning of the word distinctive to the end of the word gables” in a spoken commentary.

The requirement for a document model is that the duration of the instance is known or stated as unpredictable. This can be stated explicitly or deduced from relations with other parts of the document structure.

Temporal scaling of an instance

In order to satisfy temporal constraints that derive the duration of an instance, or satisfy constraints, individual media items need to be scaled in the temporal dimension. If the duration is to become longer then the instance can either be played slower or can be repeated until the specified duration is reached. If the duration is to become shorter, then the instance can be played faster or can be cut short.[4]

Another form of scaling, called temporal glue, [HaRe94] and [BuZe93a], 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.

Temporal scaling can be specified explicitly for atomic and composite instances in the MET++ system, [Acke94], and is derived from the document structure in the CMIFed system, [HaRB95]. Temporal scaling should be expressible within a hypermedia document model, along with an indication of how this should be achieved.

Temporal relations

Temporal relations among instances can be defined in terms of (a) whole media items, (b) parts of media items (anchors), or (c) groupings of items. Examples of each of these are: (a) a video starts simultaneously with its audio track; (b) each word in a subtitle is highlighted when it is spoken in an audio commentary (synchronization between anchors); (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 ([Bord92], [BuZe93b], [Erfl93]) categorisation of temporal constraints, put forward by Allen [Alle83], is given in Fig. 2.9. These allow all possible combinations of temporal relations between two instances to be expressed. More complex relations can be built up out of this set. Using the three examples

4. This is similar to extent reconciliation in HyTime [ISO97b].

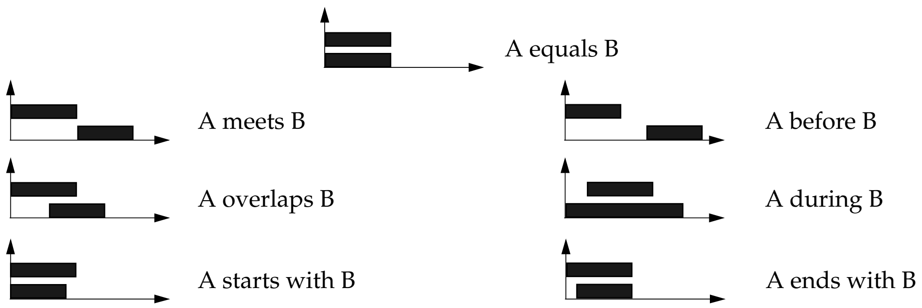
above we can illustrate the above three cases: (a) the video *starts with* the audio; (b) the highlighting of the word in the subtitle *equals* the duration of the spoken word; (c) the music *equals* the sequence of slides *meeting* each other.

Allen's relations make the assumption that the durations of the instances are known beforehand. If this is not the case, then a conditional action can be specified. For example, two media items are playing and when the first of them stops playing, which is unknown beforehand, the other one also stops [Erfl93]. Bordegoni [Bord92] gives a further categorisation of conditions as being deterministic or non-deterministic, and simple or complex conditions.

All of Allen's relations should be expressible between instances and compositions in a hypermedia document model. If, however, the duration of one of the instances in the relation is unpredictable, then a number of the relations can no longer be specified.[5]

Link transition temporal relation

When a link is interpreted as a navigation action, information needs to be specified for the presentation aspects. In particular, when moving from the source to the destination context, some specification needs to be given for the duration of



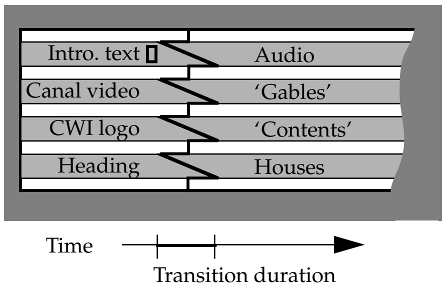
For each relation, apart from the *equals* relation, there is a corresponding B to A relation.

Figure 2.9. The Allen time relations

5. For the case that A has an unpredictable duration, the consequences for the Allen relations are the following.
A starts with B, B starts with A, B meets A, B before A: there is no change in the relationship.
A equals B, A during B, B during A, B overlaps A: the start time of B is known, but the relationship as stated cannot be guaranteed.
A meets B, *A before B*: the start time of B is unpredictable but can be scheduled when A ends.
A overlaps B, *A ends with B, B ends with A*: the start time of B cannot be scheduled since it is unknown when A will end.

Dexter layers	Model features	Model requirements
Presentation Specifications	Temporal layout	time axis, start time, duration, scaling, Allen's relations, link transition duration

TABLE 2.8. Model requirements for temporal layout



The source context fades out when the link is selected and the destination context fades in. The duration of the link transition is the time from the selection of the link to the full display of the destination context.

Figure 2.10. Temporal information in link transition

the action. For example, in the case of the source context fading out and the destination context fading in, the duration of the transition is the time from the start of the fade-out of the source context to the end of the fade-in of the destination context, Fig. 2.10. This could be more than a single duration, for example, where the source context fades out to black, there is a slight pause and then the destination context fades-in. The duration of the link transition should be expressible within a hypermedia document model.

2.2.4.2 Spatial layout

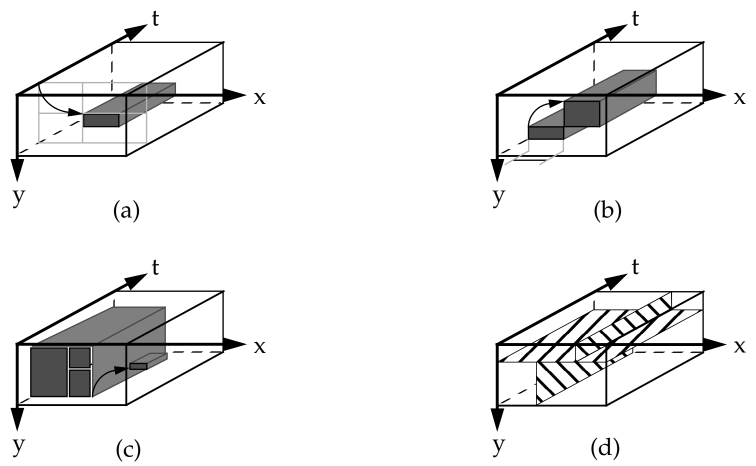
Spatial layout is an important characteristic of multimedia, defining where instances are displayed within the presentation. We give an overview of the types of spatial information that can be specified. The eight subsections address: defining a space axis in relation to which instances can be positioned, the position, extent, spatial scaling, aspect ratio, orientation and Z-order of an instance, spatial relations among instances and the spatial relation associated with a link transition.

Specification of space axis

Including a media item within a multimedia presentation requires specifying where it should be played in relation to implicit or explicit space axes. The most obvious explicit space axes are the width and height of a computer screen or a window on the screen. A space axis can be defined implicitly when the spatial extents of instances are known and these are grouped together with known spatial relations among them. We know of no systems that use this approach, however.

Position of instance
Spatial layout specifications of instances can be given in a number of ways. The most common method is to define an instance's position per item and relative to the window (or screen) in which it will be displayed, illustrated in Fig. 2.11(a). Examples of authoring systems using this approach, are Eventor [ENKY94], Mbuild [HaRe94], Authorware and Director [Macr97]. A second method is to specify the relation with respect to another instance in the presentation, illustrated in Fig. 2.11(b). A third possibility is to specify the relation with respect to a composite instance, as is illustrated in Fig. 2.11(c). Neither of these last two methods is implemented in existing systems.

Another approach, not applied in the temporal case, is to divide the available space into predefined channels[HaRB93], illustrated in Fig. 2.11(d). This is similar to specifying the position with respect to a window, but the channels have their own spatial relations with respect to each other. When the position of a channel changes then all instances assigned to that channel also have their position changed. This has the same advantage as specifying position in relation to



(a) Position is specified relative to a window.
(b) Position is specified relative to another instance.
(c) Position is specified relative to a composite instance.
(d) Channels predefine areas.

Figure 2.11. Ways of specifying position

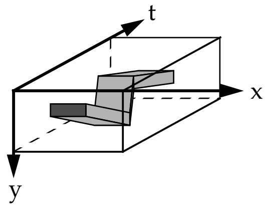


Figure 2.12. Position as a function of time.

an instance, but does not have the disadvantage that a relation has to be specified for every instance.

In all ways of defining the position of an instance, the position may vary with time, illustrated in Fig. 2.12. This is implemented in, for example, the MET^{++} [Acke94] and Eventor [ENKY94] systems.

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 instance's position within a channel can change with time. The requirement for a hypermedia document model is that the position of an instance should be expressible.

Extent of instance

As well as specifying the position of an instance, its extent also needs to be known. This may be specified explicitly implicitly or derived from relations with other instances. For example, a video has an intrinsic extent associated with its data, or a text item can be assigned an explicit extent. An example of a derived spatial extent is where a subtitle is scaled to fill the width of its associated video. In order to achieve this type of derived extent, media items need to have the property that they can be scaled.

Extent can be defined in terms of a window or channel, relative to another instance, relative to a composite instance, and can change with time. The specification of extent is required in a hypermedia document model.

Spatial scaling of an instance

When incorporating an image or video shot into a presentation the original size of the item may not be appropriate in which case a scaling operation is required[6]. This may be an absolute size, such as "200 by 100 mm", or may be relative, such as "increase to 200%". If the extent is to become larger then the instance can be enlarged or repeated to fit the specified extent. If the extent is to become smaller, then the instance can be shrunk or can be cropped. In the cases of enlarging or shrinking aspect ratio may need to be preserved, so that the specified extent may not be exactly achievable.

6. This is similar to extent reconciliation in HyTime [ISO97b].

This deformation may take place as a function of time. It may also be in relation to other instances, for example, "increase font size until heading fits above image". Spatial scaling should be expressible within a hypermedia document description.

Aspect ratio
Spatial scaling involves two dimensions, where the scaling of one dimension may differ from that of the other dimension. For images and video it may be important that the scaling factor is the same for both dimensions, i.e. that the aspect ratio be preserved. The model requirement is that the aspect ratio can be specified as needing to be preserved.

Orientation of instance
Given that there are at least two spatial dimensions within a presentation, an instance can be placed with a particular orientation at its position in the coordinate space. While this is omitted in most current multimedia systems, for applications involving three-dimensional media items, such as virtual reality, the orientation becomes more important. There is no equivalent of orientation for the temporal dimension since it is placement within a one-dimensional space. Orientation should be expressible within a hypermedia document description.

Z-order of instance
The Z-order of an instance is its position in a stack of instances occupying the same screen area. In other words, the instance with highest Z-order will be visible on the display. This is useful for occluding other instances, and also for transparent instances with graphical overlays. Z-order should be expressible within a hypermedia document description.

Spatial relations
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 those of some other media item, illustrated in Fig. 2.11(b). This would be useful for displaying, for example, subtitles next to the video they belong to, since if the position of the video is changed, the subtitles will remain in the same neighbourhood relative to the video.

Spatial relations among instances can be defined in terms of (a) whole media items, (b) parts of media items (anchors), or (c) groupings of items. Examples of each of these are: (a) subtitles are scaled to be the same width as a video; (b) a textual description is placed next to part of an image; (c) an image is scaled to form the background for a number of text elements.

Spatial constraints between instances parallel the temporal constraints shown in Fig. 2.9, but need to be combined in two dimensions. The three examples can be expressed as (a) the width of the subtitles *equals* the width of the video; (b) the text *meets* the image anchor horizontally and the image anchor *meets* the text vertically; (c) the heights of the text items *before* each other are *during* the height of the image, the width of each text item is *during* the width of the image.

Link transition spatial relation

When a link is interpreted as a navigation action, information needs to be specified for the spatial relation of the two contexts. This may be in terms of the spatial information stored in the destination context, or a specified spatial relation in terms of the source context, thus defining a spatial relationship where none existed before. This brings the destination context into the same spatial coordinate system as the source context. The position or orientation of the destination context may change with time during the transition. The spatial relation of the link transition should be expressible within a hypermedia document model.

Dexter layers	Model features	Model requirements
Presentation Specifications	Spatial layout	space axis, position, possibly changing w.r.t. time, extent, scaling, orientation, Z-ordering, link transition spatial relation

TABLE 2.9. Model requirements for spatial layout

2.2.4.3 Styles

Style information is presentation information that applies to the display characteristics of document elements. We discuss here where style information may apply and what interpretations of this may be made. Specification of style information is required in a hypermedia document model.

Media item style

The style information for a media item specifies media-related display characteristics, relevant when the system is actually presenting the item to the user. Any one display characteristic may apply to multiple media types, for example background colour or anchor highlight colour, or to only one, for example font size or line spacing.

Anchor style

The style information for an anchor is needed for specifying how the visual, or audible, characteristics of an anchor can be specified so that a user is aware that an anchor is indicating the start of a link. Examples of anchor styles are:

- a border round the anchor value;
- for text items the use of different colour or styles such as underline or italic;
- the anchor value flashes;
- the anchor value changes colour when the mouse cursor is over it;
- the anchor value "pops-up", i.e. highlights and moves position slightly when the mouse cursor is over it;

- the anchor value enlarges slightly when the cursor is over it;
- the mouse cursor shape changes when it is over the anchor value.
- For an audio item the pitch of the voice changes when the anchor value is spoken,
- or a sound effect is given just before the beginning of the anchor value.

When a user selects an anchor to follow a link, there may also be style information associated with this action. For example the source anchor may highlight (to acknowledge that the action has been registered) before the destination of the link is displayed. The destination anchor may also be highlighted briefly to distinguish it from any other anchors present in the destination context. Examples of anchor highlight styles are:

- the thickness and/or colour of a border changes;
- for text items the style and/or colour changes;
- the anchor value flashes;
- the anchor value changes colour;
- the anchor value " pops-up"briefly;
- the anchor value enlarges a small amount briefly.
- For an audio item a sound effect is given.

The style of a source anchor may depend on other properties of the link emanating from it, for example: whether the source context will disappear on traversing the link; whether the reader has already seen the destination of the link.

When a number of anchors are grouped together as the source of a link, the same style can be applied to them all. For example, when the user moves the cursor over any one of the anchors in the group all the anchors will, e.g., " pop-up".

The scenario in Fig. 2.13 illustrates a number of anchor styles.

Transition style

A *transition style* is the special effect that can be applied as a instance starts to play, fi nishes playing[SFHS91], or changes to another instance. For example, a video can " dissolve" to another video, or an audio fragment can " fade-out". Sound effects may also be part of the transition style. A transition style should not be limited to instances, but also applicable to compositions. For example, when a link is followed, the destination context can replace the source context using such effects as " wipe-down" or " zoom-in". The scenario in Fig2.13 illustrates the dissolve special effect.

Link style

This can be applied to links for representing them in network type diagrams. We do not discuss this further.

Dexter layers	Model features	Model requirements
Presentation Specifi cations	Styles	associate with anchors, instances, compositions, composite anchors and links

TABLE 2.10. Model requirements for styles

2.2.4.4 Activation state

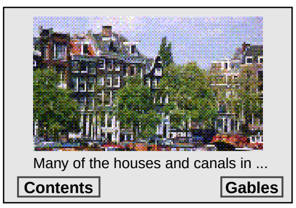
When playing a presentation different parts of the document are made active, and then are deactivated. For example, playing a multimedia sequence activates a number of media items and after the specifi ed time, deactivates them. The activation state of the items follows from their temporal ordering within the presentation. Link navigation gives the reader the choice of activating other presentations. The activation state is then dependent on the selection of links carried out by the reader.

(i) Source anchor is displayed.
Anchor style is given in instance.

(ii) User selects anchor.

(iii) Source anchor highlights.
(Anchor border thickens and
text background changes colour.)
Anchor highlight style is given in link.

(iv) Source context dissolves into
destination context.
Transition style is given in link.



(v) Destination context displayed.
(Destination anchor not highlighted.)

Veel van de huizen en ...

Figure 2.13. Anchor and transition styles on following a link

Throughout the process of playing a presentation the player software requires a record of the activation state of the parts of the document. In particular, when starting up a hypermedia presentation from its document description the player has to make a decision as to which part or parts of the presentation should be displayed initially. In the example in Fig. 2.2 the initial screen on starting up the presentation is a single contents screen. If the document describes only a single multimedia presentation, then this can be started at the beginning and played through to the end. If the document describes more than one multimedia presentation then one or more of these can be activated upon start up. The requirement for the document model is that the initial activation state of the presentation is specifi ed within the document. When the document contains links as well as multiple multimedia presentations, the requirement for the document model is that the change in the activation state of the presentation is specifi ed within the document.

In addition to activation of individual multimedia presentations, a presentation can have two states—playing and paused. This may be for a complete pr esentation, or for a single continuous media item, for example a video. In other words, a continuous media item and a multimedia presentatoin each have their own intrinsic timeline which can be traversed, or not. The normal condition when playing a continuous media item or a presentation is the “ playing”state, but it may also be in the “ paused”state, i.e. its timeline is not being traversed. The presentation is still active, however. When a link is followed from an active presentation the source context can become inactive or can remain active and either continue in its playing or paused state or change to its paused or playing state. The destination of the link becomes active, and each presentation can start up in the playing or paused state. The activation state changes on following a link are summarised in Table 2.11.

Initial state of source	New state of source	Destination
Source playing	is replaced, continues playing, or pauses	becomes active (if not already) and plays or pauses
Source paused	is replaced, starts playing, or remains paused	becomes active (if not already) and plays or pauses

TABLE 2.11. Activation state change on following a link

The requirement for the document model is that the initial playing/paused state of the presentation is specifi ed within the document. When the document contains links as well as multiple multimedia presentations, the requirement for the document model is that the change in the playing/paused state of the presentation on following a link is specifi ed within the document.

Dexter layers	Model features	Model requirements
Presentation Specifications	Activation state	initial activation state, change in activation state initial play/pause state, change in play/pause state

TABLE 2.12. Model requirements for activation state

2.2.5 Runtime layer: Interaction

Interaction with an application covers a range from sitting passively while watching a multimedia presentation to inputting different forms of information for manipulating an on-line environment. In this section we state the boundaries of the expected reader interaction with a hypermedia presentation. We assume that the reader is supported by a runtime environment for playing the presentation. Interaction features do not influence the document model, but allow aspects of the presentation to be changed by the reader at runtime.

Aspects of a hypermedia document for which interaction support can be provided by a player are the temporal flow of the presentation, the spatial layout and the links. These are not requirements for the document model itself, but a consequence of it. We specify what should be available in a runtime environment. We briefly discuss more complex interaction with an underlying application.

2.2.5.1 Temporal control

The document corresponding to a multimedia presentation contains within it information on how fast the presentation should be displayed to the reader. Controls can, however, be given to the reader to vary the speed of presentation. For example, the presentation can be halted, played from any point during the presentation, speeded up, slowed down or played in reverse. The reader can be given control over the position of the presentation on its temporal axis and the speed and direction at which the presentation is played. Implementing fast forward and reverse control may require more than straightforward implementation however [HeKo95]. Halting the presentation can also be specified from within the document as in, for example, the "pause" command in Director and the "wait" command in Authorware.

In a runtime environment basic temporal control of a playing presentation needs to be given to the reader, for example halting the presentation and continuing it.

2.2.5.2 Spatial control

Spatial control is similar to temporal control, except that the spatial axis is comparatively shorter, i.e. it is not often the case that the reader is able to see only a small part of the media item and has to scroll through the rest. Nevertheless, control can be given to the reader in the form of zooming in and out of one or more spatial dimensions, and scrolling the presentation. A more interesting form of spatial control is for three-dimensional objects where the orientation of the object is important to the reader, and a means for rotating the object in three dimensions should be supplied. Scrolling of text or large screen-based media items should be available to the reader.

2.2.5.3 Navigation control

A hypermedia document specifies links which can be expressed as navigation actions. The reader needs some means of selecting a link to follow, where the most common method is moving a mouse cursor over the desired object then selecting it. Another means is to use keyboard commands to move the cursor and make the selection. Functionality that should be supported beyond following links is recording where the reader has been and allowing the reader to return to previously seen material.

Links can be used for making choices which are similar to interactions with variables, but only in the case that the choices are finite. For example a game of noughts and crosses, otherwise known as tick-tack-toe, could be implemented using anchors and links by having a complete set of all possible stages of the game stored in the document. The reader plays the game by clicking on the desired empty square and so following the appropriate link. More complex processing than selecting from pre-authored information requires the creation of a specific application.

Application state control

A more general form of interaction can be specified between the reader and an underlying application. This allows the creation of flexible and elaborate interactions, but requires the support of input for different values of variables within an application. As an example, consider applications designed for use in an interactive learning environment. Here, support is often required for the notion of student tracking and testing. Another example is the management game described in [HaRB95], where users can interact with a simulation of a financial market via an interface implemented as a hypermedia presentation. Here the values of the variables as entered by the reader are stored in the underlying application and the results of the simulation calculations are displayed as graphs within the presentation. The requirement for a hypermedia presentation environment is that communication should be possible between it and an external application.

2.2.6 Summary of hypermedia document model requirements

In this section we discussed a number of requirements for a hypermedia document model. A summary of these is given in Table 2.13. Note that the structure of the table reflects that of Table 2.1, where the runtime layer has been omitted. This is because any aspects of the runtime layer which influence the document model are recorded in the presentation specifications.

In the next section we show that, although hypertext and multimedia document models already exist, none is suffi ciently powerful to meet all the requirements listed in this section.

Dexter layers	Model features	Model requirements
Within-component layer	Media items	temporal and spatial dimensions
Anchoring	Reference to part of media item	data-dependent specifi cation of part of media item
Storage layer	Instance of media item	reference to (part of) media item, data format, duration, start time, extent, position, aspect ratio, orientation, Z-order, style (media item, anchor, transition), start points for links, semantic attributes
	Composition	temporal and spatial composition, space/time independent composition, grouping of anchors into composite anchors
	Linking	source and destination anchor, source and destination context, transition (duration and special effect) change in activation state change in playing/paused state
	Semantic attributes	associate with anchors, instances, compositions, composite anchors and links
Presentation Specifi cations	Temporal layout	time axis, start time, duration, scaling, Allen's relations, link transition duration
	Spatial layout	space axis, position, possibly changing w.r.t. time, extent, scaling, orientation, Z-ordering, link transition spatial relation
	Styles	associate with anchors, instances, compositions, composite anchors and links
	Activation state	initial activation state, change in activation state initial play/pause state, change in play/pause state

TABLE 2.13. Summary of requirements for a hypermedia document model

2.3 Existing hypertext and multimedia models

Systems for authoring and reading hypertext material have existed for a number of years. One of the earliest was Engelbart's Augment [Niel95]. Of the more well-documented systems are Notecards ([Hala88], [Niel95]) Intermedia ([HKRC92], [Niel95]) and Hyperties [Niel95]. HyperCard [Niel95] and Guide ([OWL90], [Niel95]) are commercial systems initially available in 1985. Unfortunately, each of these systems embodies its own, implicit, model of hypertext. One of the first formal models of hypertext is the Dexter hypertext reference model, [HaSc94], which was developed by a number of hypertext system designers to express the essence of a hypertext system. This model remains the most influential in the hypertext literature.

Multimedia authoring systems have also been available, although tend to have had more interest in the commercial world, for example Director and Authorware [Macr97]. Again each system has its own, implicit, model of multimedia. The CMIF multimedia model, [BuRL91], was explicitly developed, however, in order to capture the essence of a multimedia presentation. CMIF is one of the few explicit models of multimedia.

While the Dexter and CMIF models make important contributions towards creating a full hypermedia model, neither captures all the concepts necessary. A hypermedia model should be able to describe structured information incorporating multiple continuous and static media along with temporal and spatial relationships. It should not be restricted to dealing with small multimedia presentations, but should allow the re-use of smaller presentations in the creation of larger, more complex hypermedia presentations.

We first give brief descriptions of the Dexter and the CMIF models, and then compare them with the requirements from the previous section. We demonstrate where each model is insufficient as a hypermedia document model, and propose that a hypermedia model needs to be a superset of the two. We conclude with the requirements that are missing from both models that also need to be included in a hypermedia model.

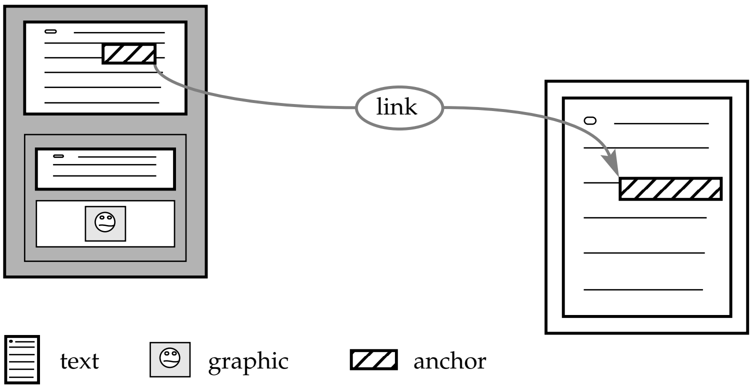
2.3.1 Dexter hypertext model

The Dexter hypertext reference model, [HaSc94], was developed as a reference model to rationalise and make explicit the concepts embedded in the then existing hypertext systems. The model defines a number of concepts present in a number of systems, but not all of which are present together in any one "Dexter compliant" system. At the time there was no existing implementation which encapsulated all of the Dexter model, although since then work has been carried out to implement the Dexter model as such, [GrTr94a]. The Dexter model is widely referenced, e.g. [Rada95] and [GrTr94b], and is the standard of comparison for hypertext systems.

The Dexter model divides a hypertext system into three layers: a *within-component layer*, where the details of the content and internal structure of the different media items are stored; the *storage layer*, where the hypertext structure is stored; and the *runtime layer*, where information used for presenting the hypertext is stored and user interaction is handled. The Dexter model describes the storage layer in detail, and it is this layer which is most relevant to a hypermedia model and of which we give a brief description here.

The Dexter model introduces atomic, composite and link components, and anchors. Atomic and composite components are related to each other via link components, where anchors specify the location of the ends of the links. This is shown schematically in Fig. 2.14.

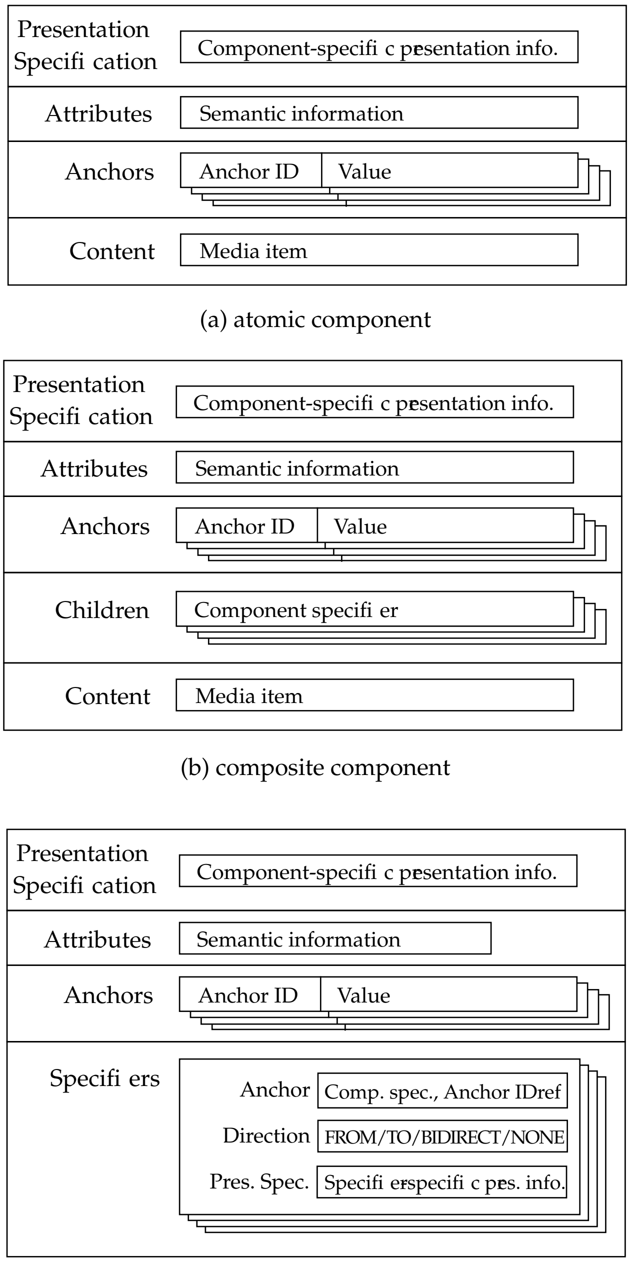
The components of the Dexter model are illustrated in detail in Fig. 2.15. A Dexter component has its own unique identifier, not shown in Fig. 2.15. A reference to a component can be made directly to its unique identifier or via a more general component specification, shown as “Component specifier” in Fig. 2.15. The latter requires a resolver function to “resolve” it to a unique identifier, for instance, to allow the addressing of a component by means of an SQL database query. A WWW URL, for example, can be a unique identifier or a component specification[7].



A composite component (left) is linked to an atomic component (right).
(There is no representation of time in the figure.)

Figure 2.14. Dexter components schematic

7. World Wide Web Uniform Resource Locator. A unique identifier is e.g. `<http://www.w3.org/Addressing/URL/>`, a component specification is e.g. `<http://www.altavista.digital.com/cgi-bin/query?pg=q&what=web&kl=XX&q=Hypermedia+Model>`.



(a) atomic component

(b) composite component

(c) link component

Figure 2.15. Dexter model

Atomic component

An *atomic component* contains 4 parts—presentation specification, attributes, a list of anchors and content, Fig. 2.15(a) (inspired by Fig. 4 of [HaSc94]).

- The *presentation specification* holds a description of how the component should be displayed by the system, the form of which is beyond the scope of the model.
- The *attributes* allow a semantic description of the component to be recorded. The form of this semantic information is beyond the scope of the model.
- An *anchor* is composed of an anchor identifier and a data-dependent anchor value. The anchor identifier is unique within a component and allows the anchor to be referred to from a link component. The anchor value specifies a part of the content of the atomic component. The anchor value is the only place in the model where the data type of the content is required.
- The *content* is a media item of a single data type.

Composite component

A *composite component* is the same as an atomic component with, in addition, a list of child components, Fig. 2.15(b)[8]. It is a collection of other components (atomic, composite or link) which can then be treated as a single component. This structuring of components is restricted to a directed, acyclic graph (by definition)[9]. The anchors of a composite component refer to the content of that component.

Link

A *link* is a connection among two or more components. Its structure is the same as an atomic component with a list of specifiers replacing the content, Fig. 2.15(c). A *specifier* defines an end-point of the link. It consists of an anchor, a direction and a presentation specification.

- The anchor refers to an anchor identifier in a component, shown as “IDef” in Fig. 2.15(c).
- The direction is one of FROM, TO, BIDIRECT or NONE. FROM and TO specify that the end-point referred to (via the anchor) is a source or destination of the link respectively. BIDIRECT allows the end-point to be both source and destination, and NONE allows neither.
- The presentation specification refers to the anchor style at an end-point of the link. A single link component can allow the expression of a range of link complexities, including a simple one-source, one-destination, uni-directional link (for example links in HTML), or a far more complex multi-source, multi-destination, bi-directional link.

---

8. A composite thus also includes content, although in the original Z specification, in the NIST publication of the model [HaSc94], a composite does not include content.
9. Although in the original Z specification, in the NIST publication of the model [HaSc94], the composition was restricted to a hierarchy.

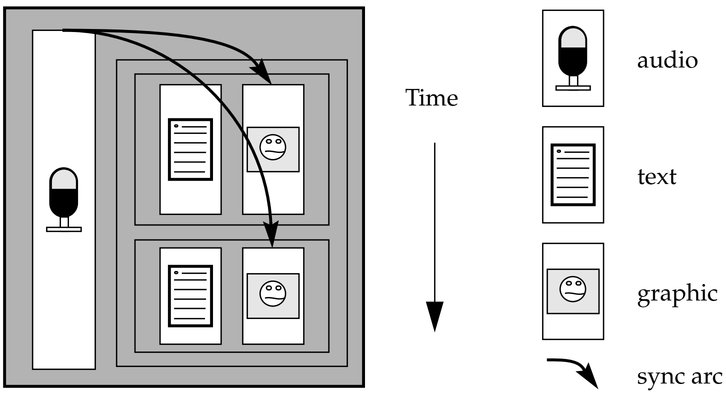
Note that atomic and composite components do not store a list of links which refer to them, and have only a list of their own anchors. It is left to the runtime layer to calculate which anchors are linked to within the scope of the hypertext environment. Anchors were introduced as a means of keeping links free from data-dependent issues, i.e. as a means of separating the storage and within-component layers. Note that since links can have anchors, links are able to link links. This has been implemented in [GrTr94a].

2.3.2 CMIF multimedia model

The CWI Multimedia Interchange Format (CMIF) model [BuRL91], describes a model for representing and manipulating multimedia documents. Multimedia document authoring systems have received scant treatment in the academic literature, until the introduction of the ACM conference on multimedia. The CMIF model is one of the first, and still one of the few multimedia document models.

The CMIF model captures the time-based organisation of information, and allows the construction of larger documents out of smaller documents (hierarchical structuring). The resulting document is a continuous presentation.

Important parts of the model, for this discussion, are that documents are compositions of atomic and composite nodes, an atomic node is presented via a channel, and synchronization arcs can specify timing constraints between two atomic nodes, Fig. 2.16.



A parallel composite component has two children: an audio atomic component and a sequential composite. The sequential composite has two parallel composite children each containing a text and a graphic atomic component. Both graphic atomic components have their starting times constrained by synchronization arcs.

Figure 2.16. CMIF components schematic

Atomic node
An *atomic node*, Fig. 2.17(a), consists of a presentation specifi cation, attributes and content.

- The presentation specifi cation is in essence the same as in the Dexter model, but is split into two parts. The fi rst part, the channel, specifi es via which channel the node will be played, and the second, the *event descriptor*, specifi es the presentation of one instance of the content. The latter can supplement or override the channel information.
- The *data descriptor* is a set of attributes describing the semantics of the data block (the same as the attributes in the Dexter model).
- The *content* is the same as in Dexter, specifi ed as a media item. This is the smallest unit that can be mapped onto a channel for presentation.

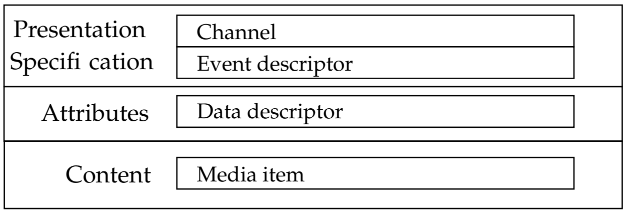
Channel
A channel is an abstract output device for playing events. This may be, for example, a window on the screen, or audio output. The channel includes default presentation information, for example font and style for a text channel, or volume for an audio channel. The only means of playing content is via a channel. The number of channels within a document is not restricted. When a document is played the channels are mapped onto physical output devices.
The channel is a means for specifying multimedia styles, in particular media item styles.

Composite node
A *composite node*, Fig. 2.17(b), is a (strictly) hierarchical composition of atomic and composite nodes. The *type* specifi es whether the composition isSEQUENTIAL or PARALLEL. In a SEQUENTIAL composite the child nodes are played one after the other, in a PARALLEL composite they start at the same time.

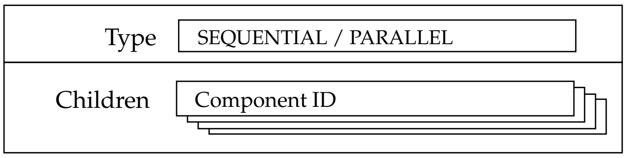
Synchronization arc
A synchronization arc, Fig. 2.17(c), specifi es system architecture-independent timing constraints between two atomic nodes. The *source* is the atomic node at the beginning of the arc and the *destination* that at the end. The *scheduling interval* gives the delay that should occur between the presentation of the two ends of the arc. This consists of the mid-point scheduling time, that is the desired delay, and allows variations to account for possible system delays when playing the presentation. The *synchronization type* specifi es how the interval should be interpreted (START, END, OFFSET) and how important it is that the synchronization condition should be met.

2.3.3 Comparing Dexter and CMIF with the model requirements

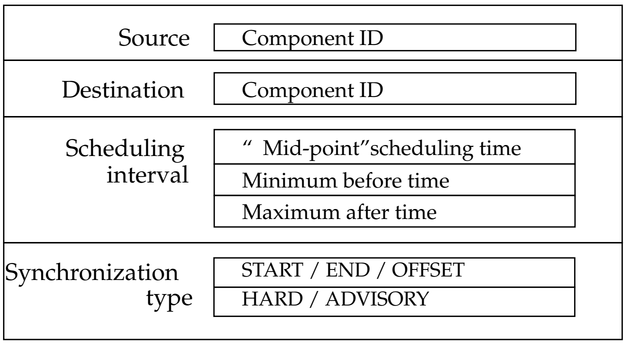
Having given descriptions of the Dexter and CMIF models we compare them with the requirements derived in Section 2.2. This is summarised in Table 2.14.



(a) CMIF atomic node



(b) CMIF composite node



(c) synchronization arc

Figure 2.17. CMIF model

2.3.3.1 Limitations of Dexter

The Dexter model allows the composition of hierarchical structures and the specification of links among components. Deficiencies of the model for expressing structured hypermedia information incorporating multiple continuous and static media along with temporal and spatial relationships are the following.

- The main drawback of the model is that while it accommodates the inclusion of continuous media items at the within-component layer it does not include time at the structuring level of documents—the storage layer. This

Dexter layers	Document model features	Requirements	Dexter	CMIF
Within-component layer	Media items	temporal and spatial dimensions	no	yes
Anchoring	Reference to part of media item	data-dependent specifi cation of part of media item	yes	no
Storage layer	Instance of media item	reference to media item, data format, duration and start time, extent, position, aspect ratio, orientation, Z-order style (media item, anchor, transition), start points for links, semantic attributes	yes no no no yes yes	yes yes partly partly no yes
	Composition	space/time dependent composition, space/time independent composition, grouping of anchors into composite anchors	no yes no	yes no no
	Linking	source and destination anchor, source and destination context, transition (duration and special effect)	yes no no	no no no
	Semantic attributes	associate with anchors, instances and compositions	yes	yes
Presentation Specifi cations	Temporal layout	time axis, start time, duration, scaling, Allen's relations, link transition duration	no no no no no no	yes yes yes no yes no
	Spatial layout	space axis, position, extent, spatial relations, scaling, orientation, Z-ordering, link transition spatial relation	no no no no no no no no	yes yes yes no yes no no no
	Styles: media item, anchor, transition	associate with anchors, instances, composites and links	partly	partly
	Activation state	initial activation state	no	yes

TABLE 2.14. Comparison of Dexter and CMIF with requirements

is the difference between temporal and atemporal composition discussed in Section 2.2.3.2.

- A second major drawback is that a link end is specifi ed only by an anchor and includes no larger information context for the link end-point, as discussed in Section 2.2.3.4. Nor does it include a specifi cation of what happens to the source context of the link when a link is followed.[10]
- Composite anchors are not explicitly included in the model.[11]

- A Dexter composite includes content, Fig. 2.15(b). This means that the presentation specifi cations (and other component attributes) apply ambiguously to either the data in the current component or all the descendants of the component.
- Dexter anchors have no explicit semantic information associated with them, i.e. no equivalent of a component's attributes.

2.3.3.2 Limitations of CMIF

The CMIF model allows the hierarchical composition of static and continuous media into time-dependent presentations. Defi ciencies of the model for expressing structured hypermedia information incorporating multiple continuous and static media along with temporal and spatial relationships are the following.

- The major drawback of the model is that it does not include links and anchors.
- The model does not incorporate atemporal composition.
- Anchor attributes are not included in the model.

2.3.4 Requirements missing from both Dexter and CMIF

A number of requirements for a hypermedia document model are in neither the Dexter nor the CMIF models. The following are a consequence of the combination of multiple synchronized media items with links among these synchronized groups.

- Both temporal and atemporal composition are required within a single model. Atemporal composition is required for providing choices which can be linked to. Temporal composition is required for synchronizing multiple items.
- Activation state information is needed, both for the play/pause state of any single temporal composition and for the specifi cation of how the activation state of the presentation changes on following a link.
- Context for linkends is needed for specifying the scope of the ends of a link.
- Transition information for a link is required, since on traversing a link the presentation as perceived by a reader should remain a continuously playing multimedia presentation.

Other requirements missing from both models are the following.

---

10. A leave/replace attribute is also necessary in the Dexter model, but this was omitted. Whether the source component of the link was left on the screen or was replaced was often implicit in the system implementation, for example Notecards [Hala88] and Hypercard [Niel95]. It could also be tied to the link type, for example in Guide [OWL90], [Niel95] a note button left the source context and displayed the destination context in a separate window, whereas the inquiry button cleared the source context.
11. Dexter *could* allow this by using the value fi eld of an anchor to specify a (list of) ComponentID/AnchorID pair(s), but this was not the original intention of the model.

- Part of a media item should be specifi able as the content of an atomic component to allow re-use of data, in particular for large media types such as video.
- Attributes for anchors are needed because although a media item is an atomic entity in a computer system (the smallest amount of data that can be retrieved from a data store) it is not necessarily a single real-world object. A single media item may portray a number of real-world objects, so that attributes on anchors are necessary for labelling them individually.
- Composition of anchors is needed for grouping together semantically equivalent parts of different media items.

2.4 Conclusion

We began this chapter by introducing an example of a hypermedia presentation. In order to create a hypermedia document model capable of expressing this type of hypermedia presentation, we need to satisfy the requirements stated in Section 2.2. Our summarised list of requirements for a full hypermedia document model is the following.

- The specifi cation of a media item needs to include information about the dimensionality of the media item, in particular whether it has intrinsic spatial or temporal extents.
- Data dependencies of the media item should be encapsulated in an instance, along with other properties describing the media item.
- Composition of components should include space/time dependent composition for the creation of continuous multimedia presentations composed of synchronized subcomponents, as well as space/time independent composition for the grouping of separate presentations among which a reader can navigate.
- Anchors are required for the localisation of data-dependent specifi cations of parts of media items.
- Links are necessary for the specifi cation of relationships among multimedia presentations. They differ from hypertext links in two ways. Firstly, given that multiple media items are collected together in a presentation, a link context is necessary to specify the scope of the linkend. Secondly, because of the temporal nature of the presentation, a pause/play attribute is necessary for the case that the source context is not replaced and for the destination context.
- Temporal layout is required for synchronizing multiple components into one continuous presentation.
- Spatial layout is required for determining where the media items are to be displayed on the screen.

- Styles for media items, anchors and transitions are required for specifying different presentation aspects of parts of media items, instances, compositions and links.
- Attributes are not required for the specification of a single presentation, but are required for more content-based creation of presentations and retrieval of already created presentations.

While Dexter and CMIF are adequate models for describing hypertext and multimedia, respectively, we have shown that both lack aspects for a complete hypermedia model. Some superset of these two models is thus needed in order to be able to describe synchronized multiple media including links. Of the requirements for a hypermedia document model, those that are not already part of the Dexter or the CMIF model are:

- context for linkends, required because of the multiple components collected together in the source and destination contexts;
- transition information for a link, since on traversing a link the presentation should remain a continuous presentation as perceived by the reader;
- attributes for anchors, since these portray basic real-world objects, in contrast with a media item which is a basic system object.

While these are our requirements for a complete hypermedia document model, a model that satisfies them also has to remain sufficiently simple to be implementable, while remaining sufficiently expressive for the majority of documents. A model satisfying these requirements has to find an appropriate compromise between these two contradictory requirements. Our own solution to defining a particular model is given in the following chapter.