5 Authoring Requirements for Hypermedia

In this chapter we discuss authoring requirements for hypermedia. We go through the document model defined in chapter 3 and discuss ways of editing and visualizing its parts. For each document component we present a list of required editing functionalities and give suggestions on how these may be visualized to aid the author. We do not go into design details of a complete user interface, nor do we attempt to combine all the different functionalities into a complete and unified system.

5.1 Introduction

This chapter states the requirements for the next generation of hypermedia authoring systems. We approach this task systematically and state our requirements with argumentation. There is an analogy with word processing, where, while initial systems were diverse, most current systems show great similarities—apparently the available features have reached equilibrium with the requirements. We wish to achieve the same for hypermedia authoring.

A hypermedia authoring system, as an instance of an interactive system, has the components model, view and controller [Burb92]. The model is a document model such as that proposed in chapter 3, the controls are the transactions we wish to carry out on the model, and the views are the ways of visualizing the model to the user. The correspondence between the model and the controls has to be a mutual cover; that is, all the elements in the document model have to be editable, and every editing action should be captured in one or more parts of the document model. Table 5.1 of this chapter, in the appendix, lists the document model elements from chapter 3 along with the sections in this chapter in which they are discussed. We concentrate in this chapter on the control which is needed over the parts of the document model and on the views that can be provided for simplifying the understanding of the state of the model. Where appropriate we discuss specific visualizations for some of the required controls.

The editing environment is divided into 4 layers, Fig. 5.1: the data layer, component layer, document layer and resource layer. These are similar to the Dexter layers [HaSc94], where the data layer corresponds to the within-component layer, and the component layer to the storage layer. The document layer requires the same input as the runtime layer, but it is a static view of the components it refers to. Calculations, such as the overall timing of the presentation, can be carried out but without actually having to play the presentation. The resource layer is perpendicular to these other layers, in that it contains information external to the component and document layers but can be referred to from them. It is a generalization of the presentation specifications in Dexter The layers communicate with each other as indicated by the arrows in the figure. We discuss each of these layers in its own section. Note that the runtime engine is not an editor as such, but can communicate with the document editor and the component layer.

While the text is rather extensive, the essence of the chapter can be found in the discussions at the end of each section and, because of the importance of timing and spatial layout in multimedia, in the timing and spatial layout sections of the document layer section. Conclusions are summarised at the end of the chapter.

5.2 Data Layer

The data layer, shown as the lowest layer in Fig. 5.1, contains the data resources external to the document itself. The store of media items, while logically in one place in the figure, may be a distributed store. This is the source of the data that forms the basis of the final presentation.

5.2.1 Media items

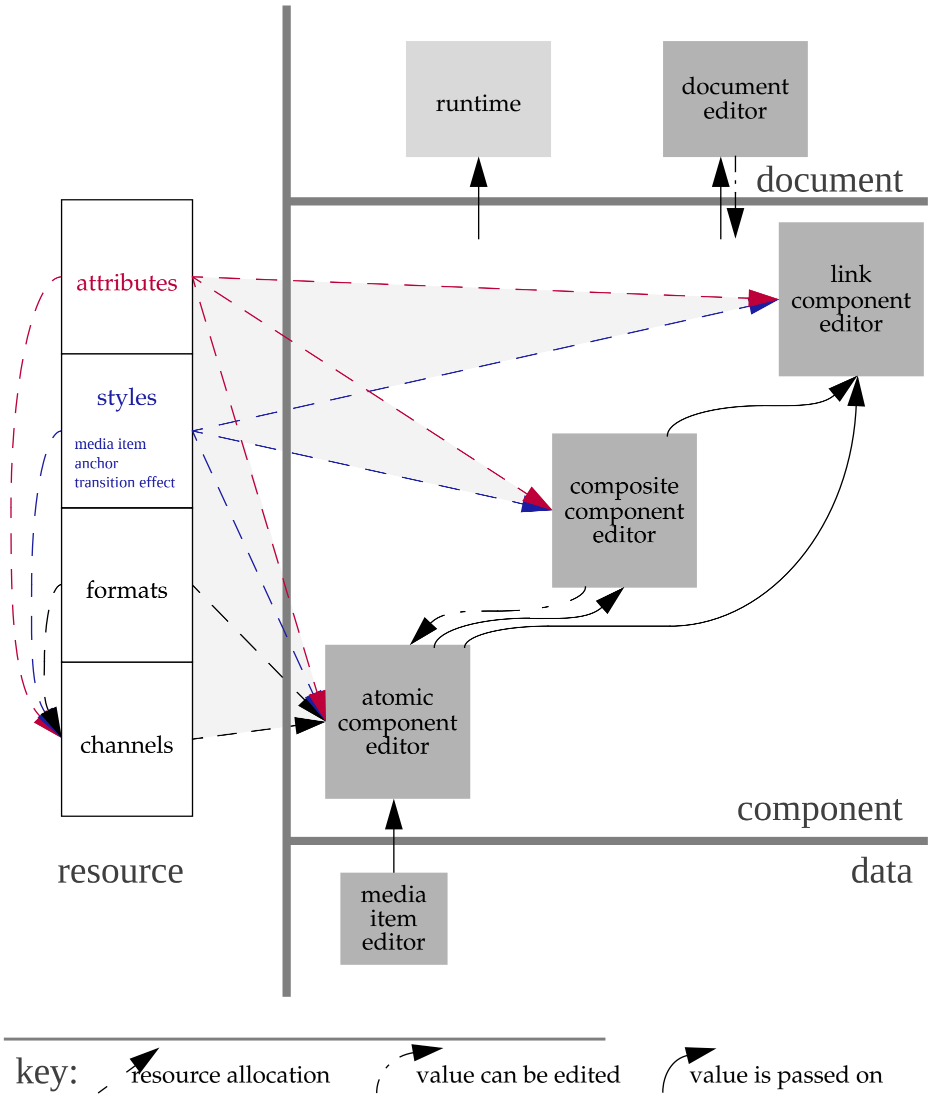
A media item is an amount of data that can be retrieved as a single object from a store of data objects, or is generated as the output from an external process. Media items are the building blocks of a hypermedia presentation—without these there are no objects to display. They can be any of the following media types: text, image, audio, moving image, combined audio and moving image. They may be derived from, e.g., a simulation, or may be three-dimensional, for example virtual reality.

Media items are of interest in the authoring process only in so far as the author needs to select one or more for inclusion in the presentation. It is an essential choice to treat media items of different origin as equivalent within the context of hypermedia authoring. We discuss here the selection of a media item and the specification of some part of it, in other words a reference to part of the media item.

Create media item

No restriction need be placed on the data types that can be processed by an authoring system. The only requirement is that the system can interpret the data format directly or convert it to a format that can be interpreted. The admissible data formats are best stored in a separate resource.

The system may include its own set of internal editors to create, edit and manipulate different data formats, but, given the widely varying editing styles



The arrow from the data layer to the component layer indicates that only the atomic component can refer directly to data. The arrows between the component layer and the document layer indicate that information from every component can be passed to the runtime system and the document editor, and that the document editor is able to edit data structures within any one of the components. Note that only the atomic component uses the format and channel resources.

Figure 5.1. Overview of data flow among the four layers

of both the media type and the author, this is best left to a specialist media-specific tool. Whether the media items are created within the authoring environment or externally, we proceed under the assumption that the media items exist and can be accessed and interpreted by the authoring environment.

By defining a media item a number of characteristics are defined by the data block itself and the corresponding data format, namely the item's intrinsic duration and intrinsic size. These characteristics are medium-dependent, for example an audio item has only intrinsic duration and an image only intrinsic size. A text data format, such as ASCII, has no intrinsic size without any further font size information. If the text data format includes font information but no layout information, then it has a number of potential sizes but no predefined aspect ratio.

Select media item

In the context of authoring, media items need to be displayed in some way to allow the author to select them for inclusion in a presentation. This is done in most systems by supplying, at the level of the operating system, a list of file names in a dialog box. The problem is that the user has to deduce the contents of the file from only its name. A direct coupling of the file name with a viewer for that data format would allow the user to select a candidate file and then play it to see if it was indeed the one required. An example of an extension to this method is to display miniatures of image and video items. This aids the selection of a particular video fragment or image from the many available.

Edit media item reference

The authoring environment should facilitate the author's task of selecting the appropriate part of a media item either as the block of data to be displayed to the user, or as an anchor value. A media-dependent description of the part of the item is required. For example, in chapter 2 we discuss media-dependent ways of specifying a part of a media item. An editor capable of displaying the media item in its entirety and selecting a part of it is required. An example of a text selection editor is to allow the author to view the complete text and drag out the desired text extent, where the system records the string offset and length. Similarly, for selecting a part of an image the author should be able to view the complete picture and specify a part of it, e.g., by indicating the top-left and bottom-right coordinates with a mouse.

When the data of the item is writable, problems can occur, since it becomes less clear which part was originally specified. Specification of the content needs to be more than a data-only specification. As an illustration, if a text item is editable then contextual information should be provided, such as keywords within the desired string and neighbouring ones just outside the string.

Once the referencing has been carried out the partial media item has its own intrinsic duration and size. It is useful to be able to play the referenced part of the item to check that it is as the author specified.

Discussion
Since the data format of a media item does not play a role in the document model, it is essential to treat the media items included in a presentation as equivalent. We achieve this through the use of an atomic component. Once a media item has been included in an atomic component, the handling of it in an authoring environment is standardized. That is why we advocate the separation of the media data and its envelope—the atomic component. In addition, the atomic component allows the inclusion of the same media item as multiple instances within the presentation.

Special attention should be paid to tools supporting the selection of media items for inclusion in a presentation, in particular, by providing an overview of a large number of items.

When a media item is not custom made for inclusion in a presentation, authors will require tools for modifying the media item to suit their purpose. There are two possibilities: either a new version of the media item can be created and modified, or a part of the existing media item can be selected. Our preference is for the latter as this avoids the proliferation of nearly equal versions and the difficulties associated with intellectual property rights. Selection of part of a media item is not supported in current generation systems. As more pre-created media items become more accessible, in particular via the World Wide Web, then authors will be able to specify the URL to include the media item in their own presentation and only require tools to select part of the item.

In the context of an authoring system, to enhance the reuse of resources we perceive it as useful to store the admissible data formats in a separate resource.

5.3 Component Layer

The component layer contains the objects that an author integrates within a multimedia authoring environment. The components contain data, information about the presentation of the data and information about the data. The *raison d'être* of the components is that they are equal status objects, regardless of the media content. The components are the only means available for recording the specifications of the presentation. In other words, any aspects of the presentation which the author wishes to control must be recorded somewhere within the component layer. The components stored in this layer are atomic, composite and link. The component layer is the central layer in the model and communicates with all other three layers in Fig. 5.1.

5.3.1 Atomic components

Since the data format of a media item should not play a role in the document model, media items included in a presentation should be handled uniformly. We achieve this through the use of an atomic component.

In the previous section we discussed that references can specify a part of a media item. It is the task of the authoring system to use these references in the atomic component in the appropriate places, namely specifying the content and the anchor values, Fig. 5.2.

An authoring system requires, as a minimum, to be able to create and delete atomic components. An atomic component consists of a content, anchors, presentation specifi cation and attributes. We discuss the presentation specifi cation in three separate parts: timing, spatial layout and style.

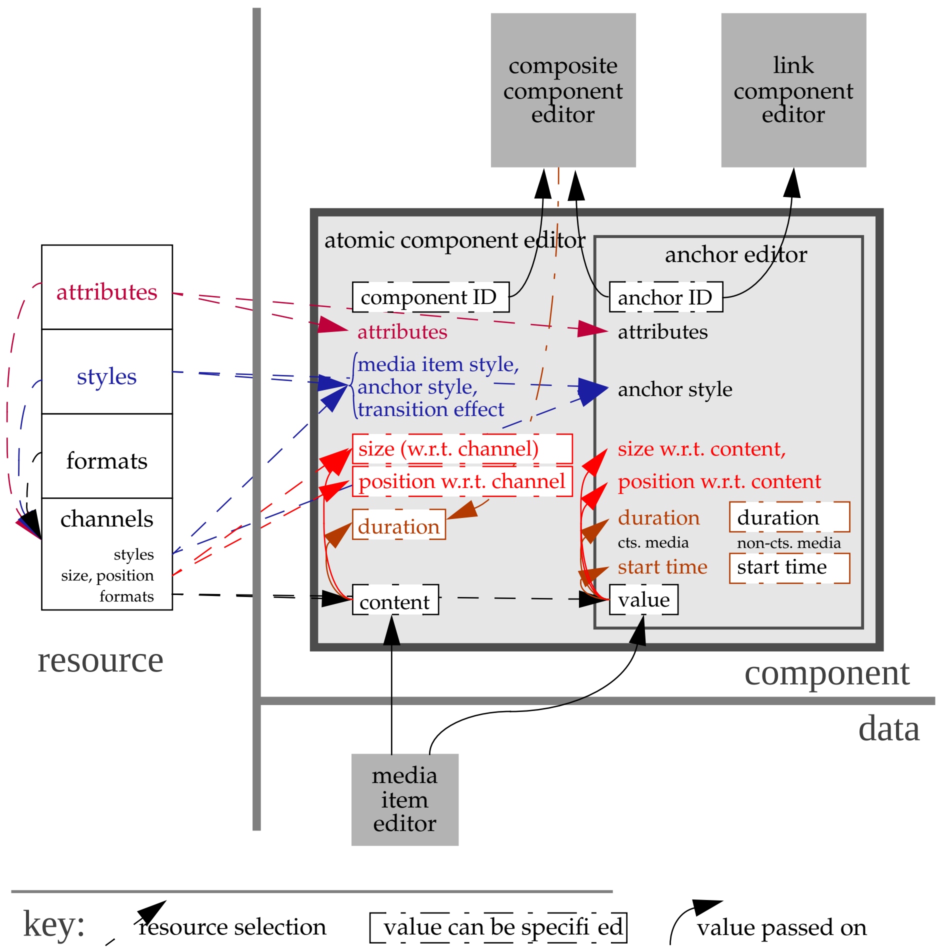


Figure 5.2. Data flow in and out of atomic component editor

Content

The content of an atomic component holds a reference to a complete or partial media item. The data which is referred to from an atomic component is always one block of data of the same media type. The data block brings with it intrinsic duration and size information which can be derived using knowledge of the data format.

The author needs to be able to specify which media item is to be used, and which part of it should appear in the final presentation.

Even though there may be only one data block, this should be selectable from a number of different data formats. At one point one may prefer a presentation in 24-bit colour rather than a 4-bit greyscale presentation. Such a switch between otherwise equal options should be decided at runtime depending on the capabilities of transport and display and of data cost. Therefore we leave open the choice of this part of the data format provided that the parameters of duration and size remain unchanged. This is to prevent the other parts of the atomic component, such as anchors and attributes, becoming invalid, and leaves the place of the data block in the presentation and its dependencies unaltered.

Anchors

Anchors are objects specifying part of a media item and can be used as the basis for creating links and synchronization arcs among components. An anchor consists of an anchor identifier, an anchor value, attributes and a presentation specification.

An anchor identifier must be specified by the author, for example labelling part of a picture with an author-selected keyword, or generated automatically by the system, for example using a generated number or picking a keyword as a name from a text item.

Anchor values should be allowed to overlap, either partially or fully. This is not always the case in current implementations, e.g. HTML. There may be an authoring requirement for changing the start-time, duration, size or position of an anchor value, with respect to the content. While this may be desirable from the author's point of view, it is impossible to supply generic tools for this since the anchor specification is media-dependent. Anchors inherit their duration, start-time, size and position from their specification with respect to the content. These properties can only be modified through scaling the complete content of the component, and thus cannot be specified per individual anchor For non-continuous media, however, the start-time and duration of the anchor can be specified explicitly

Anchor styles should be specifiable per individual anchor, which should be selectable from the same resource used for the anchor styles for the atomic component as a whole. An anchor style specified for an individual anchor overrides that specified in the atomic component or channel.

Timing

At this point, we are concerned only with the timing as specified within an atomic component. The duration of the content of the atomic component is the intrinsic duration of the media item reference. The duration of the atomic component should, however, remain editable by the author. For example, by specifying an absolute duration or one relative to the intrinsic duration of the content. Hence, the duration of the atomic component is not bound to equal the intrinsic duration of the media item reference and is stored separately. The playback system needs to resolve how the content is played to conform to the specified duration.

The timing information for an anchor is derived from the anchor value for continuous media items. For non-continuous media the start-time and duration should be specifiable by the author

Spatial layout

At this point, we are concerned only with the spatial layout as specified within an atomic component. The size of the content of the atomic component is the intrinsic size of the media item reference. The size of the component should remain editable by the author.

We consider spatial layout authoring requirements for two distinct cases. The first is that where channels are used for recording spatial information. The second is that of a model that uses no channels, and thus parallels the authoring requirements of timing somewhat closer.

For the channel case, each atomic component is assigned a channel. The author should be able to specify the position and size of the content with respect to the channel. The author should also be able to specify an absolute size or modify the intrinsic size of the content by a scale factor. The playback system needs to resolve how the content is played to conform to the specified size.

For the non-channel case, the author should be able to specify an absolute size or modify the intrinsic size of the content by a scale factor.

Styles

The styles for an atomic component should be selectable from a style resource. This allows the same styles to be applied throughout a presentation and allows changes to apply to all the components using the style. At this point, we are concerned only with the styles as specified within an atomic component. These are: media item style, anchor style and transition special effect.

The media item style is medium-dependent and includes, for example, font for text and background colour for all visual media types.

The anchor style associated with an atomic component applies to all the component's anchors and can be used for, for example, showing the position and size of the anchor value. This is a style resource although separate from that for media item styles.

The transition special effect specifies how the display of a media item is initialised or terminated, for example, "fade-in" or "checker-board". The style should be applicable to all media, but may have medium-dependent interpretations. For example "fade-in" can apply to both visual and audio media, whereas "checker-board" might use a "fade-in" for audio. This is a style resource although separate from that for media item and anchor styles.

A style specified for an individual atomic component overrides that specified in the associated channel.

Attributes
The attributes for an atomic component should be selectable from a resource of semantic attributes. Attributes should also be specifiable per individual anchor, which again should be selectable from a separately held resource. The attributes for anchors and components should be selectable from the same resource since they both describe aspects of the application domain rather than media specific or component specific information.

Discussion
The main advantages of introducing an atomic component are that it hides the data-dependencies of the media item from the rest of the environment, while also allowing the creation of complex components which are indiscriminable from atomic components. This facilitates the creation of complex presentations. We advocate authoring environments that enhance reuse and management of components.

We advocate the support for different levels of quality during presentation, as the selection should be made at runtime depending on the available display and transport resources. Specification of different measures of data quality is largely unsupported in current systems. A reason is that presentations are currently created for a specific end-user platform. We advocate authoring environments which enable the creation of platform independent presentations.

An anchor value is a data-dependent specification of part of the component's content. While it is largely an implementation issue as to how this information is stored, it does have implications for the authoring interface. If the anchor value is embedded, that is defined within the content itself, and the medium is read-only, then new anchors cannot be created. A classic example of this for the text case is HTML, where the anchor extent is defined within the text itself. If the anchor value is external, i.e. specified separately from the data but referring to it, then it needs to be transported along with the data. There would be more flexibility if both methods were possible for all media types: embedded anchors would be available for others to use, and external anchors could be created by non-owners of the material for their own use. Both approaches are supported in Microcosm [HaDH96].

The style and attributes of an atomic component should be selectable from separate resources to enhance reuse of style and attribute resources. An author should be able to assign styles on a more global basis, and can use the composite component and channel for this purpose. Attributes can be assigned at the composite level, but should be seen as complementary rather than as conflicting.

5.3.2 Composite components

Composite components facilitate the moulding of the narrative structure of a presentation while keeping the authoring process manageable even when the presentation becomes large. This is achieved by supporting the creation of larger presentations, or collections of presentations, and then allowing properties to be associated with these rather than with only the individual components.

An authoring system requires, as a minimum, to be able to create and delete composite components. A composite component consists of a list of children, anchors, a presentation specifi cation, and attributes, Fig.5.3 and Fig. 5.5. We discuss the presentation specifi cation in three separate parts: activation state, timing, spatial layout and style.

Children

An author should be able to build up a larger presentation from sub-presentations and to create groups of presentations. This is captured in a composite component by grouping together a selection of other components. These can be atomic and/or composite components[1] and the grouping can be temporal, for creating a larger presentation, or atemporal, for collecting together a number of presentations. Temporal composition requires the specifi cation of temporal relations among the children. Atemporal composition requires the specifi cation of the initial activation state for each of its children. The composition structure should be able to reflect the narrative structure of the presentation.

From the perspective of the model, a composite is merely a list of its children, but from an authoring perspective it matters a great deal how the author is able to carry out the composition. The two most obvious requirements are to allow already created components to be collected together into a composite, bottom up composition, and to allow an empty component to be created in which further components can be included, top down composition.

For an authoring environment where the compositions can be rather complex some facility is required for viewing the structure. A number of options are illustrated in Fig. 5.4. In (a) the structure is shown as a standard tree. This allows easy interpretation of the structure, but gives no immediate visualization of the timing of the individual elements nor of the presentation as a whole. In (b) the structure is shown using white title bars, where the elements under them are contained within the structure. Timing is indicated by the position and length of the elements. The structure is more diffi cult to interpret than in (a), but the flow

1. The AHM also allows the inclusion of link components in a composite component. We do not discuss the authoring aspects of this.

of the presentation is more immediately interpretable. In (c) the structure is shown using indentation. This is easy to interpret but does not facilitate the visualization of timing.

Activation state
An atemporal composite requires the specifi cation of the initial activation state of each of its children when the composite itself is played. For example, two of three children of the composite are initially active, of which one starts playing and the other is displayed but is in a paused state.

The composite as a whole has no timing information, since each of its children has its own timing information and they are not related in any temporal way. Fig. 5.5 shows the editor for an atemporal composite component.

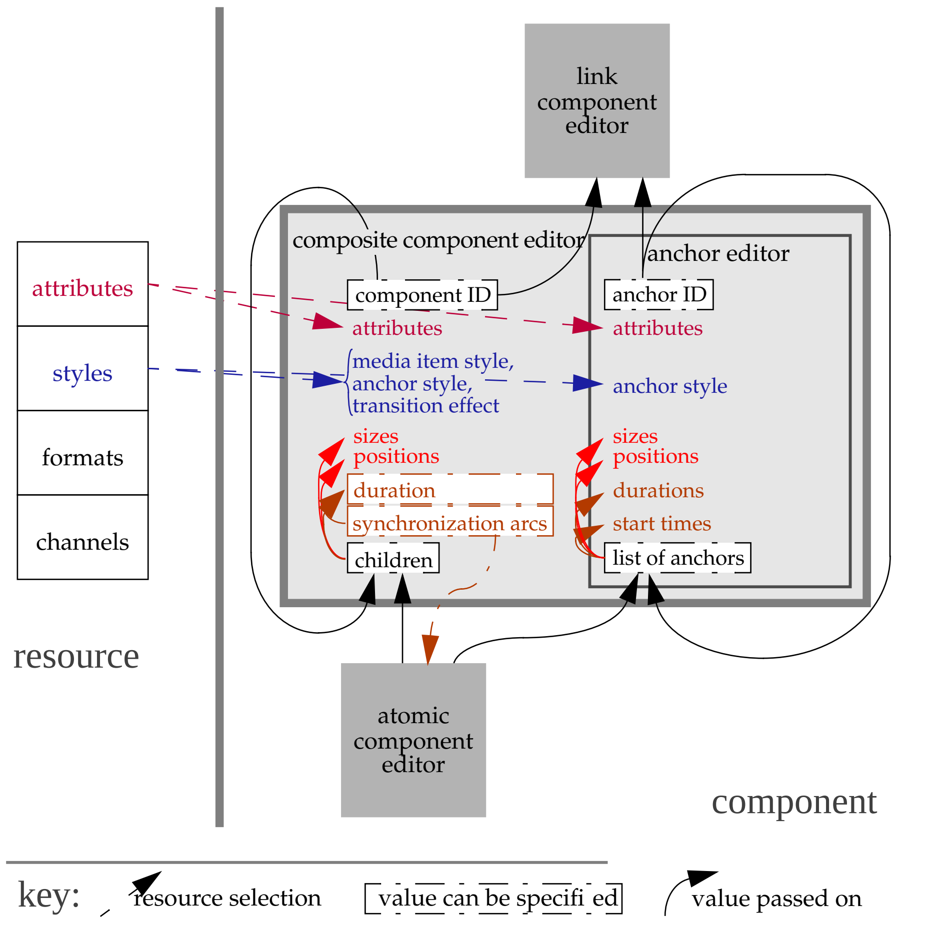
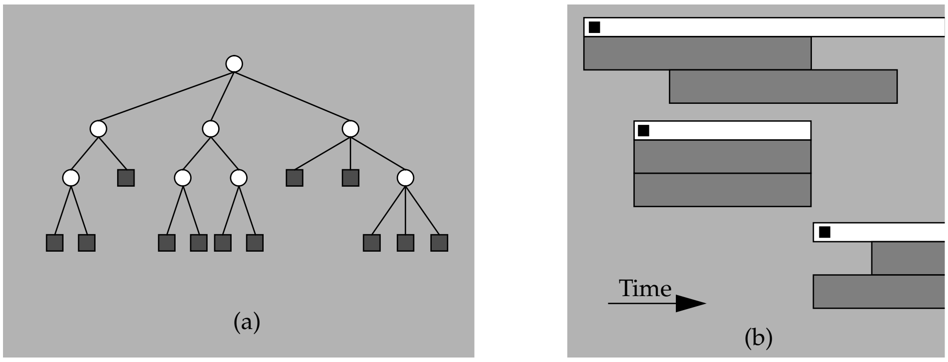


Figure 5.3. Data fl ow in and out of temporal composite component editor

Timing

An author is concerned here with how the timing of a composite component is associated with the durations of the media items and their temporal relations amongst one another. This is captured in a temporal composite including the duration and start times of all its descendants. The duration of a temporal composite component can be calculated on the basis of the durations of its children and any synchronization constraints that exist between them or their descendants. The duration of the composite component should, however, remain editable by the author. For example, by specifying an absolute duration or modifying the total calculated duration of its descendants by a scale factor. Hence, the duration of the composite component is not bound to equal the calculated duration and may be stored separately. It is left to the playback environ-




```
Scene Introduction
{
    audio object1 starts at time 0
    {    image object1 starts at time (1 for 5) at position (centre window)
         text object1 starts at time (= image object1) at position (bottom image object1)
    }
    {    image object2 starts after image object1 at position (image object1)
         text object2 starts at time (=image object2) at position (bottom image object1)
    }
    audio object2 starts at time (after audio object1)
}
```

(c)

(a) Depth shows level of hierarchy. Time is not represented.
(b) A white title bar represents structure containing grey boxes and other structures. Time is represented as flowing from left to right.
(c) Indentation shows level of hierarchy. Time is not represented.

Figure 5.4. Visualizing Composition

ment to execute the mapping of the calculated duration to the specifi ed duration at runtime.

Within a temporal composite, synchronization constraints should be defi nable with respect to single events and collections of events (i.e. defi nable between atomic components and composite components). In order to specify a constraint the two components involved in the relation need to be specifi ed along with the appropriate timing relation.

An author should be able to specify the duration of a child component using synchronization arcs in an ancestor composite component. For example, a text item is scheduled to begin and end with an audio commentary, or a sequence of slides should last as long as the accompanying music. The duration of the text item event or slide sequence cannot be found as a specifi c duration in the atomic

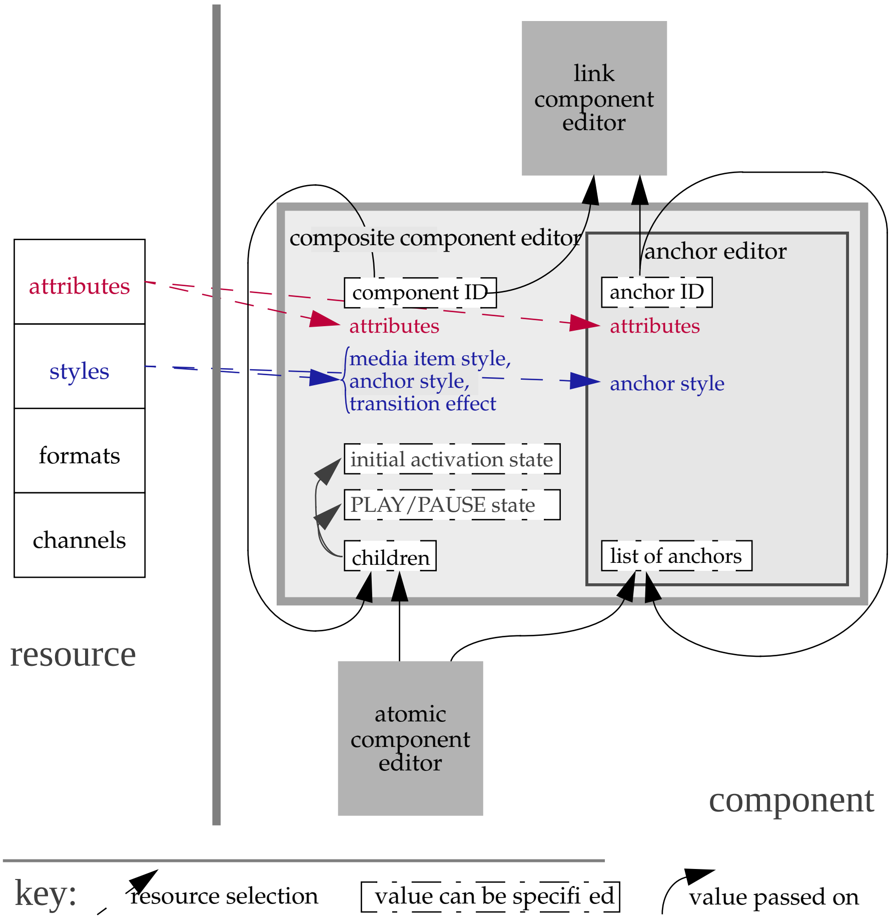


Figure 5.5. Data fl ow in and out of atemporal composite component editor

or composite component duration, but as part of the synchronization arc information in the containing composite.

Spatial layout

We discuss the channel and non-channel cases separately. In the case that channels are used, every atomic component comes with its own spatial layout. Composing these into composite components gives little flexibility in overriding any spatial information, since whatever is specified in the composite cannot override the final channel position and size. The sizes and positions of the atomic components within the channels can be changed, although the authoring requirement for specifying this via a composite component is not immediately clear.

In a model that does not use the channel construct, the size of a composite component can be calculated on the basis of the sizes of its children. Having stated this, the size of the composite may be difficult to calculate, since there is no guarantee that the children are positioned next to each other. The size of the composite component should, however, remain editable by the author, for example by specifying an absolute size or modifying the total size by a scale factor. Hence, the size of the composite component is not bound to equal the calculated size and can be stored separately. It is left to the playback environment to execute the mapping of the calculated size to the specified size at runtime.

Styles

There are three styles which should be specifiable for a composite component: media item style, anchor style and transition effects. The styles should be selectable from the same resource of styles applicable to atomic components, anchors and link components.

Styles should be specifiable at any level of the composition hierarchy. These are recorded in the corresponding composite style specification. The styles recorded in the composition structure may conflict with other styles at different levels of the hierarchy, and it is left to the player software to resolve potential conflicts.

Media-dependent styles apply to all the descendants of a composite, which may not all be of the same media type. The player software has to resolve the application of potentially inappropriate styles, e.g. by ignoring them. Note that this problem does not arise for media item styles since only one media type is involved.

Attributes

The attributes for a composite component should be selectable from a resource of semantic attributes applicable to atomic and composite components as well as to anchors.

Anchors

Anchors are used as the basis for creating links among both atomic and composite components. An anchor consists of an anchor identifier, an anchor value, attributes and a presentation specification.

Specifying an anchor identifier should be the same operation as for an atomic component. Automatically assigning a keyword as the identifier may be difficult since the anchors belonging to the composite may not use identical keywords. If the anchor refers to only one atomic anchor then the identifier could be copied.

The anchor value is given by a list of references to other atomic or composite anchors belonging to descendants of the composite. These resolve to a list of anchor identifiers in descendant atomic components. The author should be given a straightforward way of making this association.

The anchor's semantic attributes and style should be selectable from separately held resources. The attributes should be selectable from the same resource as for components. The style should be selectable from the anchor style resource, where the each anchor may have its own style. An anchor style specified for an individual anchor overrides that specified in the composite component or channel.

Discussion

In authoring a non-trivial hypermedia document the only means of reducing the complexity of the task is to support structural composition. A requirement for this is that the atomic and composite components can be treated in the same way from the author's perspective. We thus advocate systems that allow atomic and composite components to be treated equally.

A further reason for advocating composition is that the composition structure can be used by the author to reflect the narrative structure of the presentation, thus reducing the cognitive load of the author in matching the narrative structure of the presentation to the system-supported representation.

In most current authoring systems, authors are obliged to assign properties to components on an individual basis, in particular timing, styles and attributes. Composition allows the specification of properties on a more global level, relieving the author of repetitive work. We advocate the specification of timing, styles and attributes for composite components.

Composite anchors have not been previously defined, since they only become necessary where links can be created among compositions of multiple data types. In this case, the semantics of the message crosses the media boundaries. The composite anchor construct allows anchors of different data types to be collected together in a structure that the author is able to perceive and manipulate as a whole. We advocate the specification of composite anchors.

Having created a composite component, e.g. synchronizing subtitles with a video, it is useful to be able to include this component in different places in a presentation. This requires not only reuse of data (made possible by referring to the same media items from different atomic components), but also reuse of the complete composite. The work described in [GaMP94] shows that there is a need for reusing the same composite components in different situations. It is our opinion that authoring support for reuse should be provided.

Composition is perhaps the least commonly exploited aspect in current generation authoring systems. While we have described a number of desired facilities here, as more familiarity is gained with the use of composition in authoring there are likely to be more extensions in the future. An important example is the visualization and navigation of the composition structure, which although a fundamental requirement, is largely unsupported in existing systems.

5.3.3 Link components

Links express relationships among structures and are typically used for navigation purposes. In the previous sections on atomic and composite components we showed that a list of anchors can be made available. The task remaining for the authoring system is to associate a number of these references with each other via a link, Fig. 5.6.

An authoring system requires, as a minimum, to be able to create and delete link components. A link component consists of a presentation specification, attributes, anchors and specifiers. We discuss the presentation specification in three separate parts: timing, spatial layout and style. A specifier consists of a reference to an anchor in an atomic or composite component, a direction and a context.

From the author's perspective, the author wishes to specify when an end-user is able to select a different presentation and how the transition from the running presentation, the source, to the new presentation, the destination, should take place. The author should be able to specify the behaviour and have this recorded as part of the link's properties. In other words, the author wishes to create a continuous presentation where none existed explicitly in the composition structure. The link provides the timing and transition effect information that would otherwise have been recorded in a composition of the source and destination contexts. This virtual composition structure cannot be created beforehand, since it cannot be predicted when the end-user will choose to follow the link.

Specifiers

The specifiers are the information elements where the anchors of the components are referred to from a link. A specifier consists of an anchor reference, a direction, a context and a presentation specification. An author is most likely to want to specify at least one source and one destination specifier, otherwise there is no source or no destination of the link. For each specifier the following are required:

- Anchor reference
  The author needs to be able to select an anchor, via its associated compo-

nent, for inclusion in the link. This may be selected using an interface similar to navigating the composition structure.

- Direction
  For each anchor the author needs to specify whether it is a source, destination or both of the link. An author need not specify the link direction explicitly, but instead a link can be thought of as having source and destination specifi ers. In this case, the author need only create two lists of source and destination specifi ers.

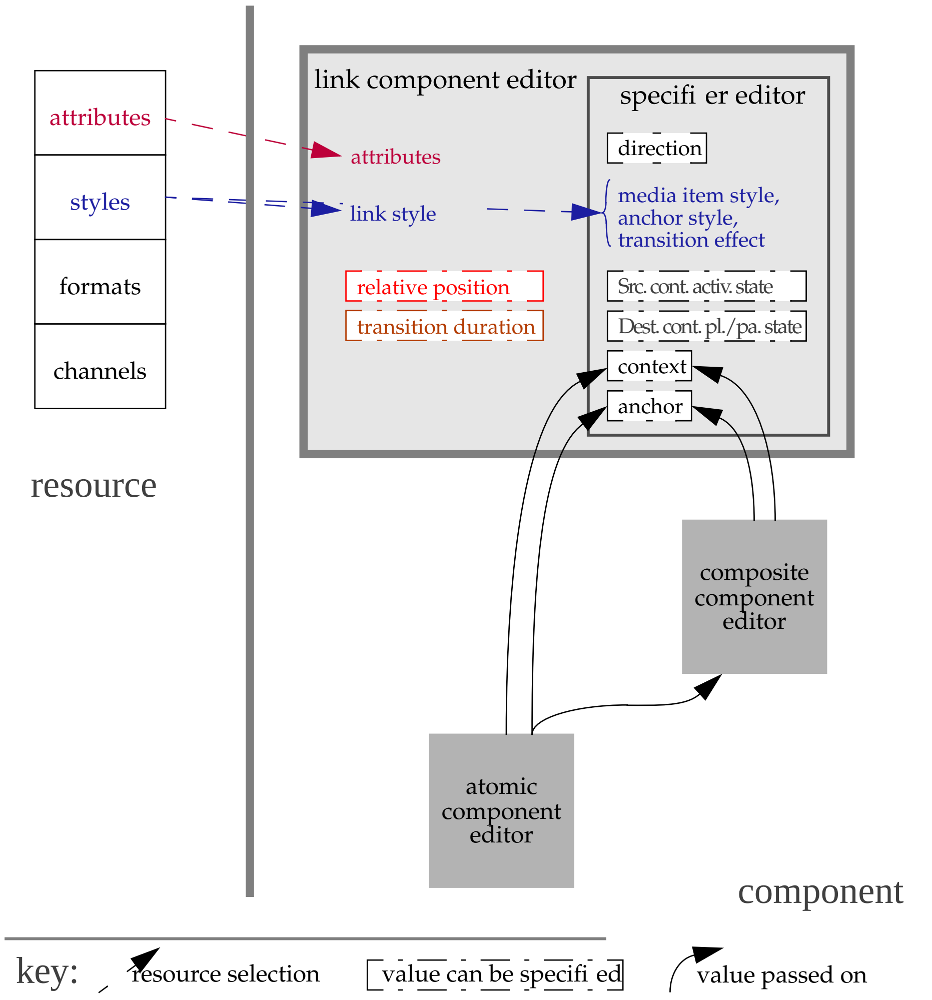


Figure 5.6. Data fl ow in and out of link component editor

- Context
  For each anchor, the author needs to specify the surrounding context of the anchor. This is given by the component containing the anchor or by an ancestor component. Selection can be supported, for example, by an interface similar to that for selecting the component containing the anchor.
- Activation state
  For each specifier which can act as the source of the link it has to be specified whether the context is paused, continued or replaced on following the link. This is the source context activation state. For each specifier which can act as the destination of the link it has to be specified whether the context is played or paused on arrival at the destination of the link. This is the destination context activation state.
- Style
  The author should be able to assign a particular style to an individual anchor, e.g. highlighting it in a particular way when leaving the source of the link. This is stored in the specifier's presentation specification. The style should be selectable from the same external resource of anchor styles as that used for the atomic and composite components. An author should also be able to assign different styles (media item, anchor or transition effect) applying to the destination context of the link, for example to preserve the visual coherence of the presentation by applying the styles used in the source context.

Timing

As part of specifying a link from source to destination context, an author needs to specify how the source context will transform into the destination context. This includes the temporal overlap of the end of the source context and the beginning of the destination context, which can be captured as the duration of the link. The start time of the destination context is valid only at playback time, since it cannot be predicted beforehand when the link will be followed by a user. The author can thus not specify when the transformation will occur.

Spatial layout

Where the destination context has no pre-defined position on the screen, the author should be able to specify its position with respect to the source context. This can be stored in the relative position of the link.

Styles

The style should be selectable from an external resource of styles defining transition effects. These allow the author to describe how the source context transforms into the destination context. This may clash with transition effects assigned to individual components, in which case the author should be able to specify which style should be given priority.

Attributes

The attributes should be selectable from a resource of semantic attributes. Although this should be the same resource as for atomic and composite components, it is likely that the attributes assigned to atomic and composite components describe objects or groups of objects and those assigned to links describe relationships. These allow the author to describe the relationship between the anchors at the source and destination of the link, for example, " is part of", " is a", " is an example of".

Anchor

We do not discuss links to links in the context of multimedia.

Discussion

The main advantage of introducing a link component is that it allows the author to specify multiple routes for the end-user to follow through a multimedia presentation.

While there is a large amount of experience with creating links in hypertext there is almost no explicit treatment of links within multimedia. Editing links within multimedia requires the management of source and destination contexts along with transition information, in addition to the usual source anchor and destination component often seen in hypertext. As yet there are no editors which support the selection of source and destination contexts or transition information explicitly. We believe that such editors need to be implemented. The first attempts to supply such support are likely to be cumbersome, and real use by authors is needed to find useful short-cuts to specifying all the required parts.

Rather than requiring an author to create links on an individual basis, a higher level approach can be achieved by creating links among abstractions instead of among concrete components. An example is given in [HoSA89], although without a description of the facility, and another example is implemented as generic links in Microcosm [FHHD90], [HaDH96]. The design of a system to help automate the authoring process, including the generation of links, is given in [WBHT97]. We advocate authoring support for high-level creation of links.

Although link specifiers support the specification of bidirectional links, in our experience it is rare to use these in multimedia. This is because of the asymmetric relation between source and destination: the user can select only one anchor at the source, whereas the destination can be a complete scene consisting of a number of media items. In the case that the destination of a link is a complete scene with no highlighted anchor, following a link back from the composite component is not possible since no visualization of the anchor is available. Although hypertext links can usefully be bidirectional, we believe this is not necessarily the case for multimedia. While we do not wish to suggest that bidirectional links have no place in multimedia, we do not demand that these be supported explicitly in an authoring environment.

The source and destination contexts of a link require a temporal relation for the case that the link is followed. We state that there is a requirement for specifying the duration of a link.

The styles and attributes applicable to a link component should be selectable from separate resources to enhance reuse of style and attribute resources. We advocate the support of specifying relevant styles for the source and destination of a link and assigning attributes to the link as a whole.

In summary, linking in hypermedia is similar to linking in hypertext, in that it allows the specification of a relation between source and destination. In hypermedia, however, it is more than this, requiring the specification of how the presentation will behave when moving from the source to the destination of the link. The author should be supported in specifying parts of the link, and in being given an overview of how the link will behave on traversal. Where possible, higher-level creation of links should be supported.

5.4 Document Layer

In the previous section we discussed timing and spatial layout in terms of an author's requirements for the atomic, composite and link components. We now discuss authoring requirements from the perspective of the document layer. The document layer provides a view onto the component layer allowing the author to deal with single aspects of the presentation using specialist editors, in particular the presentation's timing, spatial layout and link management. The results from these editors are not recorded in separate data structures but as part of the information stored in the components. While it is possible to create a hypermedia presentation using only the component editors, these do not give the author any insights into important aspects of the presentation, such as temporal and spatial layout. We advocate aiding the author by supporting the assignment of this information at the document level.

5.4.1 Timing

In the section on components we pointed out where different aspects of timing are to be found within the component layer. We confine ourselves here to the timing of the document, showing how the different timing specifications determine the timing of the presentation as a whole. This depends on the durations of events and the timing relationships among events. These in turn are based on the durations of atomic components and how these are combined into composite components. While most timing constraints in a presentation are specified beforehand by the author, following a link causes the destination context of the link to be started-up at a non-predictable time. While this cannot, and should not, be prespecified by the author, the duration of the transition can be specified.

We discuss the requirements of timing specifications, including the duration and start time of an event, synchronization relationships and higher-level editing actions. In order to facilitate the specifi cation of the timing of the events making up the presentation it is useful, if not essential, to provide an editable visualization. The temporal layout of a presentation is best visualized by means of a timeline. This allows the author to see which events occur when throughout the presentation. We illustrate the authoring requirements with example visualizations where appropriate.

Duration of event

The event associated with an atomic component has a duration. As discussed in section 5.3.1, this can be the intrinsic duration of the content, specifi ed by the author to be a relative or an absolute duration, or, as stated in section 5.3.2, determined from the synchronization arcs in the surrounding composition structure. The author should be able to specify the duration using any of these methods, see what the duration is and which method was used to specify it. Fig. 5.7 shows a typical timeline showing the durations of events and which nodes have intrinsic durations.

Start time of event

The event derived from an atomic component has of itself no explicit start time. This is captured in the composition structure via the synchronization arcs. For an atemporal composite the children are not related in any temporal way, so that the start time of any child can be determined only at run time. A temporal composite or an atomic child of an atemporal composite is a presentation. The synchronization arcs and the durations of the atomic components determine the timing of the associated children and thus the start times of the individual events.

The start times, and thus the synchronization constraints, need to be specifi able and viewable. A timeline is the ideal way of showing constraints between events in the presentation, Fig. 5.8(a). Another way of viewing is required for constraints between structures, for example as in Fig. 5.8(b) and [Acke94].

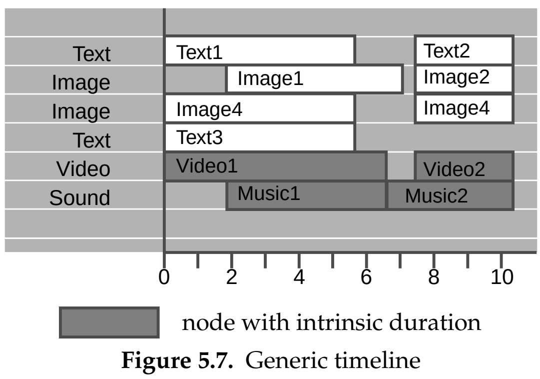


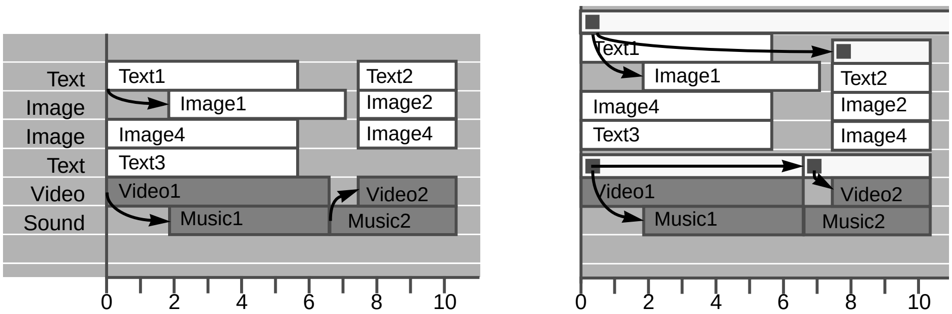
Figure 5.7. Generic timeline

Derive timing constraints from structure
In providing editing facilities for timing we seek to alleviate the author of as much trivial work as possible. Timing does not necessarily have to be defined for each individual event, but can be derived from the surrounding structure. For example, if an author groups items to be displayed at the same time they could start and finish simultaneously by default, allowing overrides to be specified as required. Examples of deriving timing from structural constraints can be seen in [Acke94] and [HaRe94]. In combination with using durations derived from the media items themselves, the author need only be obliged to specify the duration of a collection of non-continuous media.

Duration of a link transition
An author needs to be able to specify the duration of following a link from the source context of a link to the destination context. A timeline view containing only the source and destination contexts allows the duration of the transition to be visualized and specified without affecting any other timing.

Tempo
As well as specifying the durations of events, the tempo, i.e. the rate at which the presentation is displayed, should be specifiable. This is not an editing action on the presentation itself, but on the rate at which it is played.[2]

Where the tempo of the presentation can be changed, some way of indicating this is needed. There are two possibilities for incorporating this in a timeline representation: i) preserve the scale of the timeline and vary the length of the events, as illustrated in Fig. 5.9(a), or ii) change the scale of the timeline and preserve the length of the events, as illustrated in Fig. 5.9(b). A third possibility is used in



(a) Constraints between events

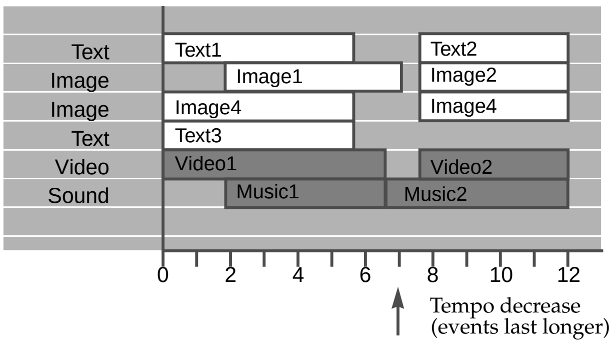
(b) Constraints from and between structures

Figure 5.8. Synchronization Arcs

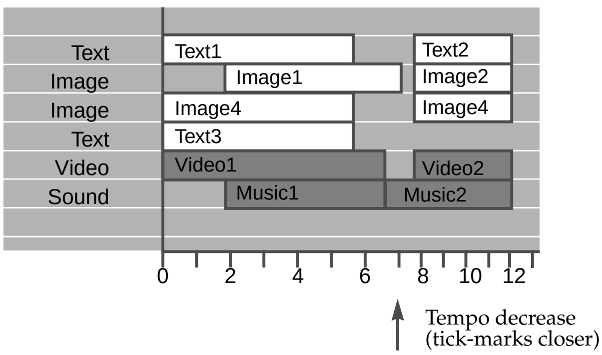
2. Note that we are unable to store this as part of the AHM.

music scores where changes to the tempo are specifi ed but not visualized, so that neither the length of the timeline nor the length of the event is changed. In a visualization for multimedia it should be left to the author to choose the representation.

Applying temporal transformations throughout hierarchy
While minimum authoring requirements can be met by supporting the specifi cation of all parts of a presentation, we strive to reduce the authoring effort. One means of achieving this is to use the composition structure of a presentation for delimiting boundaries over which a particular operation is carried out. For example, in [Acke94] a temporal transformation can be applied to a composite structure and the effect is propagated throughout its descendants. This allows several durations to be edited using only one action.



(a) Extent of events changes



(b) Scale of timeline changes

Figure 5.9. Changing tempo

Synchronization specification

Synchronization arcs allow the individual specification of timing constraints, which can be cumbersome when a large number are required. Where many similar operations need to be carried out, a considerable amount of authoring work can be saved by providing higher-level operations. Tools should be provided to support the author in specifying the constraints at a higher-level—the results of which are still stored as individual synchronization arcs.

An example is to highlight individual words in a written text as each is spoken in a commentary [HaKe97]. Anchors can be specified, in both text and audio versions, corresponding to each word. Synchronization arcs are needed to specify that the duration of the highlighting of the word is to last as long as the audio fragment. Two synchronization arcs are required per word. Authoring effort can be spared if the text and audio components are selected and a command such as "match length of anchors" is carried out. An example at a different level in the document structure is where a selection of audio fragments are to be played for the duration of a number of video clips. Each audio fragment should be faded out if too long, or repeated if too short. The two containing composite components could be selected and a command such as "match length of nodes" could be carried out.

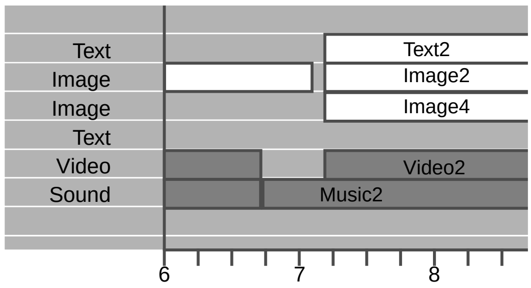
Timeline Navigation

The timeline for a single presentation may become long and unmanageable so that there needs to be some way of changing the scale of the timeline. Possible visualizations are shown in Fig. 5.10, for example in (a), a simple zoomed in view of part of the timeline, or in (b), a fish-eye view

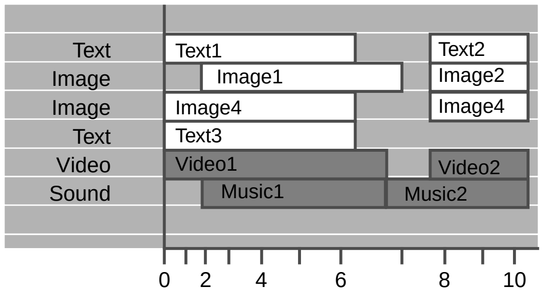
Discussion

At the document level, timing is a crucial aspect in authoring a hypermedia presentation. The complications of timing include the intrinsic dynamic aspects of the media items, the bandwidth needed for delivering them at sufficiently high quality, and the difficulties of executing temporal specifications. The timing of a presentation is so important that this is often used as the authoring paradigm, see for example the timeline based systems described in chapter 4. While this gives a useful overview of the timing of the presentation, it is not necessarily the best overall authoring paradigm. We advocate a structured approach to authoring with an associated time-based view.

Timing of individual events can be specified by allocating individual start and end times, or by using timing constraints between events. We advocate the latter, since it reduces the authoring burden when making changes to the presentation. An example of specification of timing via constraints is implemented in Firefly [BuZe93]. The system was not designed to be a complete multimedia authoring system so that while they have a representation for constraints, this is not translated into a timeline representation. We propose that both should be available to the author.



(a) Timeline zoomed in (4x)



(b) Fisheye view

Figure 5.10. Navigating timeline

Constraints should also be specifiable between groups of events. This requires the composition structure and time to be displayed in the same view. An example of this is shown in [Acke94], although temporal constraints were not implemented in the system. We propose that such a facility be available to authors.

We advocate the derivation of event start time and duration from structure as much as possible. This saves the author work by not having to specify timing details for every event.

We advocate the provision of authoring tools for specifying timing constraints at a level above that of single synchronization arcs. This saves the author work by creating multiple synchronization relations with a single command.

Explicit transitions between source and destination contexts of a link are not yet supported in authoring systems. While we discussed a single duration of a link transition, in the case that there are multiple source and destination contexts for the link there is an authoring requirement for specifying the duration of the special effect for each context separately. There is currently little experience in using explicit link transitions, and so it may be that this is more complex than needed for most cases.

In summary, the timing of events in a multimedia presentation is the key characteristic of a multimedia presentation. It should not, however, be used as the basis for authoring but, as far as possible, be derived from a structural representation.

5.4.2 Spatial layout

In Section 5.3 on components we pointed out where different aspects of layout are to be found within the component layer. We confine ourselves here to the spatial layout of the document, showing how the different layout specifications determine the layout of the presentation as a whole. This depends on the layout of events and the spatial relationships among events. These in turn are based on the layouts of atomic components and how these are combined into composite components. We now bring these together and show how they determine the layout of the presentation as a whole.

There are two possible approaches to specifying position for atomic components, either through assigning a channel to each atomic component, or by specifying position information via the composite components. For the purpose of specifying authoring requirements we discuss both approaches. When a link is followed the spatial overlap between the source and destination contexts can be specified.

We discuss the requirements of layout specifications, including the size and position of an event, how these vary as a function of time, and higher level editing actions. In order to facilitate the specification of the layout of the events making up the presentation it is useful, if not essential, to provide an editable visualization. Whether the layout is pre-specified via channels, or edited per event, is irrelevant for the display of the layout of the events. We illustrate the authoring requirements with example visualizations where appropriate.

Size of event

The event associated with an atomic component has a size. As discussed in Section 5.3.1 this can be the intrinsic size of the content, can be specified as a relative or an absolute size by the author or can be derived from an associated channel. The author should be able to specify size in relation to other objects, for example, increase the font size of a heading until it is the same width as an image[3]. Visualizing the size of a single event is trivial—the media item just has to be displayed on the screen.

Position of event

The event derived from an atomic component has of itself no explicit position. When channels are used, an atomic component has an associated channel which specifies an area in relation to another channel or window. Position information

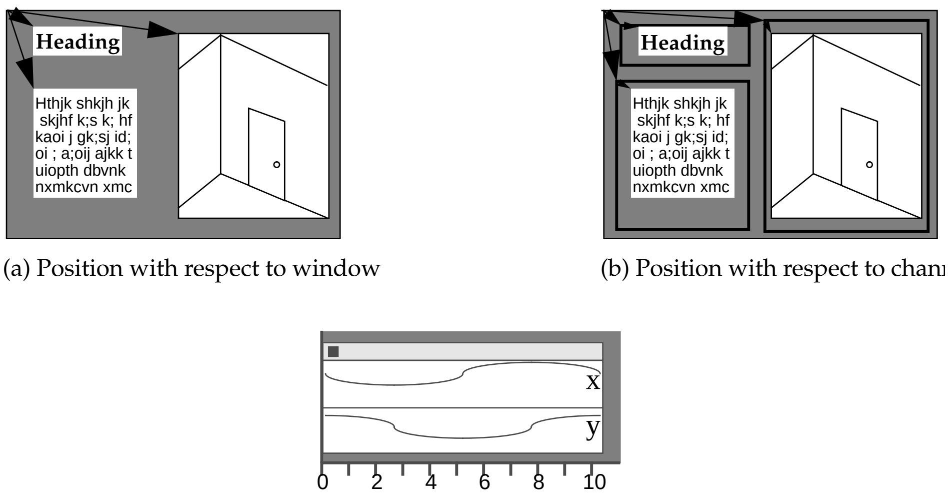
3. Note that while an author may wish to define position or size in terms of constraints, the AHM has no spatial equivalent of a synchronization arc.

for the content is specified in relation to the channel. If channels are not used, then position information can be captured in the presentation specification of a containing composite component. The author needs to be able to state which event an event should be placed relative to, and what the relative position is, for example, a subtitle is constrained to be placed at the bottom of a video and centred with respect to it[3].

While specifying the position of a single event can be done in a straightforward manner (e.g. by placing the object in the correct position on the screen, typing in coordinates or assigning it to a channel), the author is aided in this placement process by being able to see the positions of the other currently playing objects.

Visualizing the size and position of one event is trivial. Providing a visual overview of the layout of events playing at any one time is a minimum requirement, shown for example in Fig. 5.11(a) and (b). The author also needs to have some overview of the layout of the complete presentation. The layout of a presentation can be visualized by running the presentation, allowing the author to see where objects are displayed with respect to one another throughout the presentation. This method, however, is not a good way of getting an overview of the layout throughout the presentation. Visualizing the position of the objects with respect to time requires three dimensions. The three dimensions can be projected back to two [OgHK90].

All position information may vary with time. When an object's position is a function of time this is referred to as the path of the object. This can be specified



(a) Position with respect to window

(b) Position with respect to chan

(c) Position changing with time

Figure 5.11. Spatial layout

by a start point, the trajectory to be followed and the, possibly varying, speed of following the trajectory. Alternatively, it could be given by specifying the position of the object at each point in time. The path traced out by an object should be able to be visualized during authoring as well as at runtime. For example, the movement of an object over time can be visualized as horizontal and vertical positions along a timeline, Fig. 5.11(c), [Acke94].

An author should be able to specify the position of the events in a presentation while being able to view the other currently playing events and to be able to see how the layout of the presentation changes with time. An author also has to be able to specify the position of events changing over time.

Position of transition

Although transitions are normally experienced as temporal transformations, the author should also be able to specify a spatial relationship (chapter 7 of [Bufo94]), e.g. two overlapping images could blend into one another in their area of overlap. The overlap of the source and destination contexts could be specified using a spatial constraint between the source context and the destination context. This is probably most useful for the author when specified as part of a link transition dialog. On the other hand, the spatial overlap may not be specified explicitly but taken as the spatial placement already specified for each source and destination context.

A layout view containing only the source and destination contexts allows the spatial layout of the transition to be specified without affecting any other layout. The visualization would be similar to that in Fig. 5.11(a), where there would be only source and destination contexts, removed from their containing composite components, displayed within the same spatial coordinate system.

Channels

Rather than having to specify the position for every event in the presentation, some method of specifying layout information at a higher level saves the author work. Channels are a means of doing this through pre-defining areas in a window. Rather than having to specify the position for each individual event, the author need only assign the event to a channel. If the size or position of the channel is changed then the change applies to all items displayed via that channel. The use of channels also makes creating a consistent layout easier because events assigned to a channel are displayed at the same, approximate, position. The channel supplies more than a default position, since it constrains that the event be displayed within the channel boundary. Authors should be able to specify how the size and position of the event relates to the channel as follows.

- The position of the event in relation to the channel, for example, centred at the top.
- Whether the event is scaled to the size of the channel or retains the dimensions specified by the atomic component. The latter is undesirable for retaining the presentation's scalability. The advantage of the former is that,

regardless of the stored size, all the events played via the channel will appear the same size.

- If the event is scaled, whether it retains its aspect ratio. For text this is less important, but for images they will become distorted.

Channels themselves should be able to be grouped into a *layout*—a template containing a number of channels. Layouts should be able to be included in other layouts, thus providing a hierarchical structure to the layout of the presentation. Channels should have their extents and/or positions defined with respect to one another or with respect to a layout.

Derive spatial layout from structure

The author can be spared effort by specifying editing actions at a high level. While deriving temporal information from structure was an example of this, it is more difficult to state how the spatial information can be derived. If the structure is used for foreground and background objects (for example placing different pieces of text on top of a picture) then the lower levels of structure are displayed on top of the higher levels. As far as we are aware, there is no system which supports the derivation of spatial layout, and we can supply no arguments for providing such a facility. This may be an indication that solving the constraints in two spatial dimensions is so much more complex that it is better left to the author.

A combination of using structure together with a channel layout could be made, so that components specified at a high level in the hierarchy, e.g. a heading or a background, are assigned particular channels from the layout.

Applying spatial transformations throughout hierarchy

The structure of a presentation can be used for delimiting boundaries over which a particular operation is valid. In the spatial case, a transformation can be applied to a composite structure and the effect propagated throughout its descendants. For example, a composition of a video with its accompanying subtitles could be scaled to fit a new window This type of transformation has not been implemented as far as we are aware.

Where channels are used, these transformations should be applied to the layout structure, captured by the hierarchy of channels, rather than the document composition structure.

Spatial layout navigation

In the case of timing, an author needs to be able to visualize the overall timing of the presentation as well as detailed synchronizations between events. In the case of layout, however, the author needs to compare different layouts with each other. For example, when one picture is replaced by another the replacement should appear exactly in the same place on the screen. In both examples it is actually the timeline that the author requires to navigate, and not the layout itself. The requirement for the author is not that they should be able to navigate one particular layout, but that the presentation can be viewed at (at least) two points on the timeline at once in order to compare layouts.

Discussion

Spatial layout and timing are both extremely important aspects in authoring a hypermedia presentation. While the complications of timing are to do with the intrinsic dynamic aspects of the media items and the difficulties of carrying out the temporal specifications, spatial considerations are complicated by their two-dimensional nature. Not only are the sizes and positions required in two dimensions, but there is no screen-dimension left for providing an overview to the author. Other means then need to be found.

In many current authoring systems, in particular the structure-based, flowchart and script systems discussed in chapter 4, position per event is specified independently of any other events. This makes it difficult to get the positioning correct without playing the result. A minimum authoring requirement is that the author is able to see the positions of co-occurring events while editing the position of an event. A time-line based view, where the position of the event is editable, is a means of providing this functionality.

In all the systems discussed in chapter 4, the position of an item is defined with respect to a containing window or the screen. This has as consequence that spatial constraints between events, or groups of events, cannot be specified. The constraint approach would allow the position, or size, of an event to be changed and have other events follow it. It is unclear as to why systems, so far, support only the specification with respect to a window. It could be that this is sufficient for most purposes. On the other hand, no system has yet provided the opportunity for authors to specify constraints with respect to other events or groups of events.

We advocate the use of channels for specifying consistent layout in a presentation. Channels were introduced into a document model for hypermedia not only for the specification of layout, but also for the grouping together of media items that use similar resources. Resource control of a document retrieved over a networked environment can thus be controlled for the complete presentation rather than for each individual media item. A more in-depth discussion of resource control lies outside the scope of this thesis.

A disadvantage of using channels for spatial layout is that spatial constraints cannot be specified directly between events. For temporal information, however, we do advocate the use of synchronization arcs directly between events. The difference between the spatial and temporal cases is based on the notion that time is typically longer than the screen is large, in other words a larger number of events can be placed along the timeline than can be placed on the screen at one time. That is, the advantage of having consistent layout for different groups of events outweighs the disadvantage that if the position of one event is changed then all the other positions need to be changed. The same action of consistently dividing the screen space into reusable units would be to divide the timeline up into equal length chunks. This is not generally useful for a presentation consisting of events of different lengths. For the temporal case, allowing constraints to define the relationships means that the effects of editing actions can be applied easily throughout the timeline without any author intervention. For the spatial case, the layout changes have easily overseeable effects with no need for constraint solving.

The position of a component can be a function of time, so that the associated media item traces out a path. Visualizing the path is important though difficult. For example, horizontal and vertical projections along a time axis do not give an intuitive feel for how an object will behave on the screen, nor how all the objects will appear in relation to one another. The only other way of comparing paths of objects is to play the presentation. We advocate the provision of a visualisation for an event's path, but are unable to recommend a particular approach.

As well as allowing the position of an object to change with time, the size of an object should also be able to change with time. Size changes over time can also apply to a transition, where, for example, as the destination context appears it expands to fill its allocated space. Neither of these possibilities is, as far as we are aware, supported in current authoring systems. We advocate the support of changing object size in an authoring environment.

Explicit transitions between source and destination contexts of a link are not yet supported in authoring systems. While we discussed the spatial relation of single source and destination contexts, it may be that there are multiple source and destination contexts, and that for each destination context there needs to be information specifying the position of its display. There is currently little experience in using explicit link transitions, and so it may turn out that this is more complex than needed for most cases.

Specifying the position of events in a multimedia presentation is, along with the timing, one of the most fundamental requirements. We have shown in our discussion that there are a number of important authoring requirements which should not be overlooked or dismissed as trivial.

5.4.3 Link management

A hypermedia author is concerned with the creation of a presentation narrative, which, while supported by composition of components, also requires the specification and maintenance of links among these components. An author is not only required to create individual links among components but also to ensure that the possible paths through the presentation are meaningful to the end-user. During the process of creation the author requires some means of ascertaining which links are not yet complete, which have been verified and which still need to be verified. A complete link maintenance environment needs to be provided for the author to carry out the various tasks required. We do not attempt in this section to provide a complete list of tools for link management, but instead provide some examples of support that should be considered for inclusion in a system.

Find incomplete links

The author is aided by providing a list of links which are syntactically incomplete, i.e. those that do not possess one or more of: source anchor, source component (implied if a source anchor has already been given), source context, transition duration and effect, destination anchor (optional), destination component (optional), and destination context. The author needs an independent list from which a link can be selected then edited to the author's satisfaction before choosing another incomplete link from the list.

Check complete links

Having created a number of links, the author requires to check them for semantic validity and presentation aspects. For the case of the link the latter are the duration of the transition and a special effect.

The author can be aided by the system providing a list of the links in the document which can be annotated with a note of whether a link is satisfactory or not, and a means of generating a list of links the author still has to check. Sublists of this sort may also be useful, for example, list all the links with the selected component as part of the destination context.

Find unlinked components

The author should be able to ask for a list of the composite components which, given the document structure (both composition and link structures), can never be reached from the initial starting point of the presentation. It should then be left to the author to act on this information.

Discussion

A hypermedia author requires to specify and maintain links among separate presentations. The author needs to check that the possible paths through the presentation are meaningful to the end-user, and whether each path is complete or still under construction. For each link the author needs to verify the source and destination contexts and the transition. When the number of presentations and links becomes large then not only the end-user but also the author can become lost in hyperspace. We advocate the provision of a link management system to aid the author in organising the creation and maintenance of links. One method of providing this functionality is to support the creation of author-specifi able lists of links which can be annotated with status information.

5.4.4 Presentation Control

Having created the various aspects of the presentation, an author needs to preview what has been created. This can be by playing all events in the presentation that are displayed at the same time, but also by viewing subsets of the events. In order to keep playback overhead to a minimum, previewing should be able to be done on any atomic or composite component, allowing complete scenes to be viewed, or only those events which are part of a local context. The author should also be able to play all events starting from a particular time, and be able to fast forward and reverse the presentation.

While the presentation is playing, the author should be able to see which parts of the editable representation are currently playing. For example, by highlighting the currently playing events in a timeline or composition hierarchy. Fig. 5.7 provides an illustration of a timeline representation. For the case of viewing a subscene some means is needed of showing that not all the events are being played. This would require a combination of structure and timeline view, e.g. Fig. 5.8(b).

Discussion

While an author requires to play a presentation to check through it, this should not be an independent compile/run cycle that requires large amounts of the author's time. A more efficient use of the author's time is to allow the author at any point in the authoring process to play any part of the presentation. While the presentation is playing, the author should be able to select any media item in it and inquire where this is included in the presentation's structure. In other words, the editing environment should be able to pass any (consistent) part of the presentation's specification to the playback system, and the playback system should be able to communicate to the authoring environment which parts of the presentation are actually playing. The author then has the most control and flexibility of when and how much to play of the presentation under construction.

5.5 Resources

The environment layer contains stores of information that, while outside the scope of an individual hypermedia document specification, are resources that are needed by a document and can be reused by multiple documents. These resources include data format, style, channel and attribute.

5.5.1 Data format

A data format resource is needed so that a playback system knows how to present the data, and also so that spatial and temporal information can be deduced from the media item for use in other parts of the system, for example display in a timeline view. SGML, and thus HyTime, for example, specify the data format using internationally recognised data descriptions. While these need to be specified, the author should be supported by the environment by recognizing standard formats as much as possible and not demanding that the author specify the format for every media item in the presentation.

Discussion

External information is needed in order to be able to interpret the data format used by a media item. This is best assigned via a resource, otherwise a universally parsable description is needed in every atomic component.

5.5.2 Styles

Style information allows the display characteristics of events to be described. The situation is similar to word processors where a number of styles can be defined and applied to all the text annotated with that style—when the style is changed the appearance of all pieces of text with that style changes. From an author's perspective, the styles are selected and applied rather than edited or created. We propose providing higher-level editing notions for assigning style information, allowing the author to apply desired changes to large groups of objects at once. We describe each of the three style types briefly before discussing their authoring requirements.

- *Media item style*
  Media item styles specify display characteristics for media items. They may apply to multiple media types, for example background colour, or to only one, for example line spacing for text. Style information for media items can be stored with the atomic component referring directly to the media item, with the channel associated with the atomic component, with an ancestor composite component, and with a link specifier (applicable to the context when it is used as a destination).
- *Anchor style*
  Anchor style information specifies display characteristics for anchor values. This may be when the anchor is displayed as part of a media item, or during the process of following a link. Style information for anchors is recorded in the presentation specification for an anchor in atomic and composite components, in channels and in the presentation specification of a link specifier (applicable to the source and destination anchors). Style information in the link specifier can be used for expressing how the source and destination anchors will be displayed when the link is followed.
- *Transition effect*
  Transition effects are the special effects which can be applied to the beginning or end of single events. These can be stored with the atomic component, with a channel associated with the atomic component, with an ancestor composite component, and with a link specifier (applicable to destination contexts). Transition effects can also be applied to the source and destination contexts when following a link. These are stored as the transition information of the link component.

The author requires to be able to assign styles for groups of objects and for overrides for single events. This gives the advantage of high-level operations, at the same time maintaining flexibility for overrides. Styles for different groupings should be able to be assigned via channels and composite components. Styles for single events should be able to be assigned via atomic components.

Each style type should be selectable from its own resource of styles. A media item style may be applicable to one or more media types. The anchor and transition styles, however, should be applicable to all media types. Where the same literal interpretation is not possible (for example for audio and visual media) there should be different interpretations for all media types. This allows an author to assign a style to any component without needing to know whether it is applicable.

Discussion

We advocate the use of style resources, because the more information that is included in resources, the better it is for style integrity, reuse and maintenance. These increase consistency within a presentation and facilitate applying changes at a high level. We are not aware of any system that currently allows the specification of styles for multimedia, or that allows styles to be assigned to channels or throughout a composite structure. There exist already, however, styles for paper documents and hypertext, in particular DSSSL [BuRL91], CSS1 [LiBo96] and XSL [ABCC97]. These are commonly used in text-based environments for specifying structured documents[4]. The application of styles to hypermedia can be very powerful, but also complex, as shown in initial work being carried out on style sheets for hypermedia, [OHRE97].

5.5.3 Channels

A channel collects together data type, layout and style information into one object which can be re-used by several atomic components. It can also have semantic attributes associated with it. Having defined a channel it can be re-used within a document and in multiple documents. The channel is used at playback time for resource control, but we do not discuss this further here. Where a document model uses channels, an atomic component requires a channel to be assigned to it. A channel itself also requires an editing environment.

Edit channel

While the authoring environment needs to know the precise data format for a media item, it is easier for an author to use media types. In other words, the author need only know that they are using an image or a video, and not know the actual data format used. The system should be able to work this out from the data file header information.

The position of a channel needs to be defined in terms of some base layout, for example the screen, a window or another channel.

The styles appropriate to the channel are those applicable to the atomic component: namely media item style, anchor style and transition effects. These should be selectable from the same resource as for the components.

The attributes of a channel should be selectable from an attribute resource, otherwise the semantic relationships among the attributes are unknown.

4. In particular documents using SGML style mark-up. CSS1 is the style sheet language for HTML. XSL is a simplified version of DSSSL for use on the World Wide Web.

Assign Channel
In the atomic component editor the channel associated with the component needs to be selectable from a list of appropriate channels. For example, an atomic component with an ASCII file as a media item could be assigned a text or an HTML channel. If the author has already assigned a media item to the component then the system should show only those channels appropriate for the data type. The system should show only those channels not already assigned to another atomic component playing at the same time in the same composite. The author should also be able to select a current layout.

Discussion
The channel construct allows a number of properties and resources to be allocated to groups of events. Channels themselves can also be grouped and inherit properties from the group, but use overrides as appropriate.

An author should be able to see groups of channels combined together in a layout, that is a grouping which the author perceives as useful. It is likely that a layout fills a window. Note that the composite component hierarchy is orthogonal to the layout channel hierarchy associated with the descendant atomic components. Channels or layouts belonging to a layout channel would inherit styles and transition effects from their parent layout. We advocate the use of layout channels in an authoring environment.

From an author's perspective, it is a restriction that channels have a fixed size and position. Channels are not only introduced as a high-level style/layout specification, but, at least as importantly, as virtual resources that a playback environment can use for pre-calculating screen usage, or audio channel allocation. While from an author's perspective it is useful to vary the size or position of the channel it is questionable whether this is still acceptable for the channel's virtual resource function.

In summary, the introduction of channels into a hypermedia presentation environment brings with it advantages and disadvantages. The advantages are in the area of applying and manipulating resources at a high level. The disadvantages are in the area of flexibility of layout specification. We perceive the advantages as outweighing the disadvantages, thus advocate the use of channels in an authoring and in particular the playback environment.

5.5.4 Attributes

Attributes allow semantic information to be attached directly to different parts of a hypermedia presentation. Having assigned these they can be used for media-independent information retrieval and more automatic link creation. From an authoring perspective, we are concerned with assigning attributes to the appropriate components and not editing the attributes, nor their semantics, as such. What is important is the level of detail at which attributes can be assigned. Attributes can be associated with an atomic component, each anchor of an atomic component, a link component, a composite component, each anchor of a composite component and channels.

An author should be able to assign one or more existing attributes to a particular component, or an anchor within a component. The author should be allowed to add to the list of attributes, but only in a way that can be re-used by other applications. Ideally a large thesaurus should be available, allowing searches, generalisations and specializations of concepts.

Discussion

The authoring systems discussed in chapter 4 make no mention of attributes, with the exception of the Athena Muse, [HoSA89], which gives no details on how the links among composite structures are based on high-level abstractions. The exact use of attributes, and in particular how they can be used for information retrieval and more automated generation of links, is currently a topic of research. We have carried out some initial work in this direction, see for example [HaBu95a], [WBHT97].

Current technology requires that each attribute be assigned personally by an author. This is, yet another, tedious task that we should try to automate as far as possible. Tools for, for example, showing salient stills of a video could be used for an author to point at objects and state (preferably verbally to avoid long point and click sessions in menus) what they are. The system should be able to find the boundaries of the object, and assign the same attributes whenever the object reappears later, or whenever a similar object appears (requesting confirmation from the author when this is ambiguous). Ideally, internal consistency checking should also be going on, so that the system can warn the author if different attributes are assigned to multiple occurrences of the same object.

The use of attributes within documents is not necessary for creating a hypermedia presentation. We believe, however, that with the increased use of internet technology the demand for finding objects of diferent media will increase greatly, with the consequence that assigning attributes, initially to objects and later to presentations, will become a standard feature. An authoring environment should thus provide standard facilities for choosing and assigning attributes.

5.6 Summary and Conclusion

In this section we go through the four layers of our authoring model and restate the most important conclusions. This results in a list of requirements for a next generation hypermedia authoring system.

The two main, and conflicting, goals of building an authoring system are that it is comprehensive in the data structures it is able to edit, and that the authoring effort is reduced as much as possible. The former goal prescribes a minimal functionality that has to be supported, and the latter requires an exploration of the authoring interfaces for providing the functionality. A corollary of reducing the authoring effort is that a presentation should be authored only once, but contain suffi cient information for playback on difering end-user platforms.

We divide the authoring environment into four separate but communicating layers—data, component, document and resource. This facilitates maintenance and fl exibility of associating diferent resources with the same components. The data layer shields the other layers from data-dependencies, and the document layer supports alternative views of the component layer. These different views supply the information needed for a particular task, thus focusing the attention of the author.

5.6.1 Data layer

The separation of the data layer ensures that data dependencies are hidden from the main authoring process, allowing the presentation of a uniform interface regardless of data dependent details.

Media item

While we advocate the separation of the media item and its containing atomic component, and in so doing appear to lessen the importance of a media item, it remains the basis of a presentation. The author requires tools for selecting one media item from many, and then to select the appropriate part of the chosen media item for inclusion in the presentation. The latter requires the provision of media-dependent tools for selecting part of a media item. The environment also needs access to descriptions of the data formats of the media items, which we recommend be stored as a format resource.

5.6.2 Component layer

The component layer records the specifi cations for a hypermedia pesentation by means of the atomic, composite and link components.

Atomic

The main benefi t of intoducing an atomic component is that it hides the data-dependencies of the media item from the rest of the environment. This allows complex presentations to be constructed while retaining maintainability.

To increase consistency and enhance reuse of resources, the style and attributes of an atomic component should be selectable from separate resources.

While an atomic component contains information relating to a particular media item, it is still possible to exchange equivalent items, where this does not make other properties invalid. This allows media item selection to be made at runtime, depending on the available display and transport resources. This facilitates the creation of platform independent presentations.

Composite

From an author's perspective, atomic and composite components can be treated in the same way, thus simplifying the authoring process. Composition facilitates the creation of complex presentations from other smaller presentations, and also allows the reuse of a composite in different places in a larger presentation. The system abstractions should mirror the author's tasks as closely as possible in order to reduce the complexity of the authoring process. In particular, the narrative structure of the presentation can be reflected in the composition structure. This structure should be used as much as possible for generating other detailed aspects of the presentation, in particular the timing constraints.

Composition supports the specification of properties, such as styles and attributes, on a more global level thus relieving the author of repetitive work and enhancing style consistency.

The introduction of the composite anchor allows multiple anchors to be collected together in a structure that the author is able to perceive and manipulate as a whole. This allows link-ends to cross media boundaries thus better reflecting the semantics of the link.

Link

The main advantage of introducing a link component is that it allows the author to specify multiple routes for the end-user to follow through a number of multimedia presentations. Linking in hypermedia requires the specification of how the presentation will behave when moving from the source to the destination of the link. This requires the specification of source and destination anchors, components and contexts, along with transition information. The transition information includes the duration of the link.

The styles and attributes applicable to a link component should be selectable from separate resources to enhance reuse of style and attribute resources.

Rather than requiring an author to create links on an individual basis, a higher level approach can be achieved by creating links among abstractions instead of among concrete components.

5.6.3 Document layer

While the component layer is able to record all aspects of a presentation, editing individual components is rarely the best view for constructing a presentation. The impression of the presentation as perceived by the reader is dominated by such aspects as timing and spatial layout, hence they are sufficiently important to warrant specialist authoring support. Link management support is needed for dealing with the otherwise intangible collection of links.

Timing

At the document level, timing is a crucial aspect in authoring a hypermedia presentation, so that the author should have access to an overview of the timing. Timing should not, however, be used as the basis for authoring, but follow from the composition structure where possible. Deriving the start time and duration of an event from the structure saves the author work through not having to specify the details for every event.

When the start times of an event cannot be derived from the structure, it should be specified using timing constraints between events rather than absolute start times. This reduces the authoring burden when making changes to the presentation. Constraints should also be specifiable between groups of events. To facilitate the editing of constraints between groups of events, the composition structure and time should be displayed in the same view. Authoring tools for specifying timing constraints at a level above that of single synchronization arcs should be provided where possible.

As well as the duration of the atomic and composite components, the author also needs to be able to specify the duration of links, that is the duration of the transition from the currently playing presentation to the destination of the link.

An example of specification of timing via constraints is implemented in Firefly [BuZe93]. An example of displaying the composition structure and time in the same view is given in [Acke94].

Spatial layout

Specifying the position of events in a multimedia presentation is, along with the timing, one of the most fundamental requirements. An author should be able to view the positions of co-occurring events while editing the position of an event. We recommend that position and size be specified with respect to a channel, and not with respect to another event, which facilitates consistency of layout. In most authoring systems, spatial layout is not specified via channels, but by assigning a position on the screen to the media item. This is done by positioning the item where it should appear, [ENKY94], [HTOH96], or by specifying the x and y positions over time, [Acke94]. In both cases it is difficult to get an overview of the position of the object relative to other objects or over time. A timeline view provides the author with easy access to a screen view from any point along the timeline, e.g. Director, [West93].

The position of a component can be a function of time, so that the associated media item traces out a path. We advocate the provision of a visualisation for an event's path, but are unable to recommend a particular approach. Path specification is illustrated in [West93] and [Acke94]. As well as allowing the position of an object to change with time, the size of an object should also be able to change with time. Size changes over time can also apply to a transition. Neither of these is, as far as we are aware, already implemented in a multimedia authoring system.

Link management

A hypermedia author requires to specify, verify and maintain links among separate presentations. The author needs to check that the possible paths through the presentation are meaningful to the end-user, whether these paths are complete or still under construction, and whether the transitions are appropriate. A link management system should aid the author in these tasks, as a minimum by providing author-specifiable lists of links which can be used as check-lists. A number of aspects of link management are illustrated in Microcosm, [HaDH96].

Presentation Control

Direct communication between the editing environment and the playback system should be possible during the authoring process. The editing environment should be able to pass any part of the presentation's specification to the playback system, and the playback system should be able to communicate to the authoring environment which parts of the presentation are actually playing.

5.6.4 Resource

The use of resources allows properties, such as data format, style and attributes, to be assigned on a re-usable basis and leaves them open to external standardisation. Examples of data formats are GIF, PNG [Bout96] and MPEG [Pere96], of styles CSS1 [LiBo96] and of attributes PICS [KMRT96]. Channels are a useful resource abstraction and can themselves be allocated properties from the other resources.

Data format

External information in order to be able to interpret the data format used by a media item should be available via a resource.

Style

The use of styles in hypermedia is not yet commonplace, although their application is a powerful tool allowing different styles to be applied to the same final document. Without the use of a style resource there is a lack of style consistency, hence we advocate the use of a style resource. Styles allow the appearance of a presentation to be changed with very little effort. Style information applicable to hypermedia includes media item, anchor and transition effect styles. These are three separate style resources media item, anchor and transition, each of which is applicable to the three component types.

Channel

The channel construct allows a number of other resources to be allocated to groups of events. Channels themselves can also be combined together in a layout. Channels, or layouts, inherit their resources and position from their parent layout. Any or all of these should be overrideable in a particular channel instantiation.

There is a tension in channels being used as resource allocators and their potential inflexibility in layout specification. We currently hold the view that, for the case of presentations destined to be played on multiple platform types over a network, the resource arguments override those for layout flexibility

Attributes

The use of attributes within documents is not necessary for creating a hypermedia presentation. The exact use of attributes, and in particular how they can be used for information retrieval and more automated generation of links, is currently a topic of research. We believe that the demand for finding objects of different media will increase greatly, with the consequence that assigning attributes will become a standard feature. This will only be useful if the attributes are held in a separate resource.

5.6.5 Designing a real system

This chapter describes the editing facilities required for a complete hypermedia authoring system. These on their own are insufficient for a useful design of a system, since some of the requirements may conflict. The following chapter describes the authoring system CMIFed which incorporates a large number of the listed features in a unified set of tools.

Appendix

Model elements			**References to sections**
Channel	Presentation Specification	Channel ref. Position & extent Style	5.5.3 Channels 5.4.2 Spatial layout 5.5.2 Styles (media item, anchor, transition effect)
	Attributes		5.5.4 Attributes
	Media type		5.5.1 Data format
Atomic Component	Presentation Specification	Duration Channel ref. Position Extent Style	5.4.1 Timing (duration) 5.5.3 Channels 5.4.2 Spatial layout 5.4.2 Spatial layout 5.5.2 Styles (media item, anchor, transition effect)
	Attributes		5.5.4 Attributes
	Anchors	Anchor ID Pres. Spec. Attributes Value	Specified 5.5.2 Styles (anchor) 5.5.4 Attributes 5.2.1 Media items, 5.4.1 Timing (duration, start-time) 5.4.2 Spatial layout (size, position w.r.t. content)
	Content	Media item ref. Data-dep. spec.	5.2.1 Media items 5.4.1 Timing (duration) 5.4.2 Spatial layout (size)
Temporal Composite Component	Presentation Specification	Duration Sync. arcs Style	5.4.1 Timing (duration) 5.4.1 Timing (start-time) 5.5.2 Styles (media item, anchor, transition effect)
	Attributes		5.5.4 Attributes
	Anchors	Anchor ID Pres. Spec. Attributes List of anchors	Specified 5.5.2 Styles (anchor) 5.5.4 Attributes 5.3.1 Atomic components, 5.3.2 Composite components
	Children	Comp. ref.	5.3.1 Atomic components, 5.3.2 Composite components
Atemporal Composite Component	Presentation Specification	Initial activ. state Play/pause Style	5.3.2 Composite components 5.3.2 Composite components 5.5.2 Styles (media item, anchor, transition effect)
	Attributes		As for temporal composite
	Anchors		As for temporal composite
	Children		As for temporal composite

TABLE 5.1 AHM elements and where discussed in this chapter

Model elements			**References to sections**
Link Component	Presentation Specification	Duration	5.4.1 Timing (transition duration)
		Relative position	5.4.2 Spatial layout
		Style	5.5.2 Styles (link)
	Attributes		5.5.4 Attributes
	Anchors		Not discussed
	Specifiers	Source cont. act.	5.3.3 Link components
		Dest. cont. p.p.	5.3.3 Link components
		Style	5.5.2 Styles (media item, anchor, transition effect)
		Anchor	5.3.1 Atomic components, 5.3.2 Composite components
		Context	5.3.1 Atomic components, 5.3.2 Composite components
		Direction	5.3.3 Link components

TABLE 5.1 AHM elements and where discussed in this chapter