7 Summary and Conclusions

In this chapter we first give a summary of the conclusions reached throughout the body of the thesis. We then discuss two applications of our work, the first based on the model defined in the thesis and the second on the authoring system described.

7.1 Summary

This thesis derives the requirements and defines a document model for hypermedia. The model combines synchronization relations among multiple, possibly continuous, media items along with linking structures among the components of the document. The thesis goes on to specify authoring system requirements for the model and describes the implementation of the authoring system CMIFed.

7.1.1 Model

In Chapter 2 we derived the requirements for a hypermedia model. We argued that a model sufficiently expressive to describe a hypermedia presentation is required to include the following:

- multiple media types and data formats,
- temporal and spatial layout information,
- grouping of individual and group elements,
- the ability to address part of an individual media item,
- relationships among groups or individual elements, and
- media-independent descriptions of media items.

Given the importance of temporal and spatial layout for a multimedia presentation, we described these requirements in greater detail. We also described the requirements for a document model for describing link activation, since for presentations containing multiple synchronized, continuous elements a precise way of defining the activation state of the presentation is required. The reader should also be able to control the playing of the presentation itself at run-time.

We concluded that existing hypertext and multimedia models were insufficient as models of hypermedia. Of the requirements for a hypermedia document model, those that are not already satisfied by existing models are:

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

In Chapter 3 we defined the Amsterdam Hypermedia Model (AHM) which satisfies the requirements derived in Chapter 2. In order to maintain an overview of the parts of the model, it is divided into the three Dexter layers: within-component, storage, and runtime. The storage layer communicates with the within-component layer by means of an anchoring mechanism which encapsulates data-dependent details in the within-component layer. The runtime layer communicates with the storage layer by means of presentation specifications which allow the styles of components to be stored independently of the components themselves.

The AHM incorporates the following novel extensions to the Dexter hypertext reference and the CMIF multimedia models:

- The presentation specifications within the model are explicitly stated as temporal, spatial, style and activation information. Each aspect occurs throughout the model and we demonstrated how the occurrences relate to one another. This allows all aspects to be edited for a single component, or the same aspect for multiple components.
- Media item reference, anchor reference and channel reference, in addition to a component reference, are used throughout the model. This allows components to be selected on the basis of, for example, semantic annotations.
- Content is specified explicitly as a media item reference along with a corresponding data-dependent specification, which allows different parts of a single media item to be included in multiple presentations.
- Anchors have been extended to include semantic attributes and presentation specifications, including start time and duration for an atomic anchor of a non-continuous media type. The semantic attributes allow anchors to be searchable. The presentations specifications allow different appearances to be assigned to anchors and in non-continuous media types the anchor value may be visible for only part of the duration of the component.
- Composition of anchors was introduced to allow the collection of items of similar semantics in different media, thus reducing the number of links required.
- Composition of components is of two types: temporal and atemporal. Dexter expressed only atemporal and CMIF expressed only temporal. The inclusion of both types of composition within one model allows the composition of presentations where temporal relationships are known only at runtime.

- Activation state information has been incorporated throughout the model. This includes initial activation state, play/pause state and change in activation state on following a link. Activation state information allows the initial activation state of the presentation to be recorded in the document and specifies how this changes when a user follows a link.
- Link components have been extended to include context and activation state in the link specifier. The context allows the specification of how much of the running presentation is affected on following a link. The activation state specifies the behaviour of the source and destination contexts.
- Transition information, including transition duration and special effect, has been incorporated in the model. This allows the specification of the behaviour of the presentation when a user follows a link.

While the parts of the model have been shown to be necessary for describing a hypermedia presentation they must also be shown to be sufficient. We demonstrated this by describing the models implicit in a selection of existing hypertext, multimedia and hypermedia systems in terms of the AHM.

We concluded that our objective of providing a comprehensive, yet not overly complex, model for hypermedia presentations has been satisfied by the AHM.

7.1.2 Authoring

To support the author in the creation of hypermedia documents an authoring environment is required.

Chapter 4 analysed the authoring paradigms used in a selection of existing multimedia authoring systems. The paradigms illustrate different ways of providing similar functionality in a hypermedia authoring environment. They do not, however, provide a solution to the problem of which functionality should be provided in such an authoring environment. The paradigms were analysed for their suitability for different parts of the authoring task, namely creating narrative structure, temporal information, spatial layout and links among individual presentations. Structure based systems are more suited to editing the structure of a presentation, and, where the structure reflects the temporal structure, are also useful for editing the presentation's timing. Timeline based systems are more suited to showing the timing throughout a presentation and the timing relationships among parts of a presentation. None of the paradigms discussed is particularly suitable for editing layout or for creating links, although the structure-based paradigm allows the different parts of the link to be specified.

We demonstrated how an event within a multimedia presentation can be described using each of the paradigms. We concluded that each paradigm is most suited to a particular editing task, that no single paradigm is sufficiently powerful for covering all editing aspects of a hypermedia presentation, and that several interfaces within a unified environment are required.

Chapter 5 derived the authoring requirements for a system that supports the creation of hypermedia documents conforming to the AHM. To maintain an overview of the authoring environment it is divided into 4 layers: resource, data, component and document. The resource layer is a generalisation of the Dexter presentation specifications. The data layer corresponds to the within-component layer and the component layer to the storage layer. The document layer includes both a static analysis of the aspects stored in the component layer as well as run-time aspects.

The resource layer contains the resources used for the different aspects of the document, for example a data format resource required for interpreting the data, style information for fonts, semantic attribute information dependent on the application domain, and layout information. The importance of including the resources as a separate layer is that each can be replaced by a similar resource while leaving the document structure itself unchanged. This allows multiple presentations to be generated from the same underlying document structure, e.g., layout can be tailored to specific output environments.

The component layer is supported in an authoring environment by allowing the individual editing of the components themselves. An important aspect of the component layer is that atomic and composite components have been defined so that they can be treated equivalently. This facilitates a uniform approach to inserting and deleting components in a document and enhances maintainability.

Temporal and spatial layout play a particularly important role in hypermedia presentations, and these aspects have to be coordinated among multiple elements. The document layer allows the editing of these aspects and communicates the information which requires to be stored to the component and resource layers.

As well as stating the requirements for the document layer, illustrations of potential user interfaces were given for aspects such as editing temporal and spatial information. In particular, timeline illustrations were given for showing temporal constraints, changes in tempo, and navigating the presentation timeline.

Chapter 6 described the authoring environment CMIFed and stated where it deviates from satisfying the requirements stated in Chapter 5. This demonstrated that an authoring system broadly conforming to the requirements derived in Chapter 5 can be implemented. The omissions from the implemented system were categorised according to whether they would be worthy of implementation, or, from experience, were not found to be necessary.

Our conclusions in Chapter 6 were that the AHM is not an overly complex document model, since the key parts of the model have been implemented in a working system, and that the specified functionality derived in Chapter 5 is broadly implementable.

7.2 Application of work

The work reported in this thesis has contributed to and benefited from important international collaboration. The most notable of these we report below. The first is more closely related to the model part of the thesis, the second to the authoring part.

7.2.1 SMIL

The World Wide Web Consortium [W3C97] is an international industry and research consortium which coordinates the development of specifications and reference software that are made freely available. The goal of the consortium's Working Group on Synchronized Multimedia [SYMM97] is to define a declarative document format for synchronized multimedia documents for the Web. The name of the format is SMIL—Synchronized Multimedia Integration Language—pronounced "smile" [Hosc97b]. This language provided an opportunity to test the robustness of the AHM, since the goals for the documents it should be able to describe are very similar to those for the AHM, [Hosc97a].

The model proved to be a solid base from which to work and the AHM played a major role in defining the requirements for the SMIL document model. While it was the main influence on the definition of the requirements, for pragmatic reasons, such as simplicity of use and implementation and acceptance of the first version, some of the finer details of the AHM have had to be postponed until later versions of the SMIL language. In essence, however, the SMIL model contains the same basic parts as the AHM:

- references to the basic data items using URL's (a reference to part of a data item is not yet included);
- data items are encapsulated within atomic components;
- anchor specification is possible, but is less prominent than in the AHM;
- temporal composites, of types parallel and sequential, containing atomic, parallel or sequential elements;
- a repeat attribute which allows the content of an element (atomic, parallel or sequential) to be repeated (this is captured implicitly in the AHM as a specified duration in terms of the intrinsic duration of the component, and is a particular case of satisfying the specified duration by repeating the content);
- synchronization relations (but these can be created only among siblings to alleviate the complexities of the player software);
- layout is defined in a separate structure and referred to from an atomic component;
- links can be defined among atomic or composite components.

While the AHM is in general a superset of the model implicit in SMIL, a number of runtime aspects included within SMIL are outside the scope of the AHM:

- lip synchronization, that is synchronization defined at every point during the presentation of two or more synchronized continuous media elements;
- switching among different data types at runtime.

Major aspects included within the AHM but not in SMIL are:

- atemporal composition. A SMIL presentation (in version 1.0) has a single contiguous temporal extent. It is expected that this will be included in future versions to better support linking among presentations;
- links with multiple sources and destinations;
- the association of semantic attributes with components.

The AHM proved to be sufficiently expressive that it could be used as a basis for the SMIL language. The modelling aspects that were of particular relevance were the aspects which should be addressed in the language, in other words the requirements as given in chapter 2, and the details of link descriptions. The major obstacle in proposing the AHM as a base for SMIL was its complexity. The perceived needs of users do not currently match the capabilities of the technology, so that it was difficult to explain why parts of the model were required before the more basic parts were implemented. As a result, a number of aspects of the AHM are omitted in the first version. An illustrated high-level description of the model is given in [Bult97], and the current working draft in [Hosc97b].

7.2.2 Chameleon

The Chameleon project [Cham95] is an ESPRIT-IV research and development project whose aims are to define and implement multimedia authoring and presentation tools. One of the goals of the Chameleon project is to allow the creation of a single source document which can then be (semi-)automatically adapted for playback on a range of end-user platforms. As such, Chameleon is a suitable test-bed for the robustness of the underlying AHM model and for the authoring system CMIFed built around the model.

The original UNIX versions of the CMIFed authoring environment and player formed the initial core of the project and have been ported to the Macintosh and Windows platforms. The project thus had immediate access to an authoring environment which was able to create hypermedia presentations and whose functionality, broadly speaking, satisfied the authoring requirements as stated in Chapter 5 of this thesis. In addition to the availability of the authoring environment, CMIFed has been able to contribute to the Chameleon project through the document format is supports. The document format is based on the AHM and the information in the document format is recorded explicitly, which allows it to be translated with relative ease to other formats. Target formats currently being implemented in the Chameleon project are MHEG-5 [ISO97a] and SMIL.

The Chameleon project has contributed to the development of CMIFed with regard to our analysis of the authoring environment's user interface. Within the thesis, we have been able to give analysis of the user interface based on our own experience. The Chameleon project partners, who are already experienced in creating multimedia presentations using other tools, have been using the early versions of the system and giving feedback about the design of the user interface. This work is still in progress.

The main contributions of the CMIFed environment is that it supports a rich hypermedia document model. As a result, the complete environment has been able to be tailored to the SMIL document format with comparatively little implementation effort.

7.3 Discussion and future work

7.3.1 Extensions to the AHM

The AHM provides a sufficient model for hypermedia and forms a robust and well-formed model. The categorisations within the model of components, spatial layout, time, semantics, styles and linking behaviour would not need to be changed. The model could, however, be improved or extended within each of the different aspects, e.g. by separating the spatial layout hierarchy from the channel element, or by including a timeline object within a temporal composite.

The model could be extended further to include more runtime aspects. For example, by including a way of selecting among synchronized streams of information, or auto-firing of links.

An extension to the model which may require changes beyond the detailed level is the inclusion of absolute time.

7.3.2 Facilitating authoring

To alleviate the author of tedious, and time-consuming, work, we have been investigating the potential of automating the authoring process by generating presentations from higher-level semantic descriptions. Initial work in this direction has already been carried out on two aspects. The first, reported in [WBHT97], is on the design of a system that integrates a store of media items annotated with semantic attributes with a means of making selections from them and combining them into a presentation that can be displayed to an end-user. The second is on the integration of a standard reference model for the process of creating dynamically generated multimedia presentations with a document model for hypermedia, in this case the AHM [HaWB98]. Further work needs to be carried out on the problem of relating pieces of content with semantic descriptions of the content, either semi-automatically or by hand.

A different approach is being taken for a similar goal, that of generating presentations suitable for different platforms, which relates to work being undertaken in the Chameleon project. The approach is to encode the CMIF format in HyTime [ISO97b], thus making the hypermedia semantics of the document format explicit. In order to express runtime aspects of a hypermedia presentation, extensions to HyTime are required [ROHB97b]. Given these, a hypermedia document expressed in HyTime can be converted to other document types, or to particular output formats, using standard tools. These include the declarative specifi cations of transformations using style sheets. Existing style sheets are, however, text based so that work needs to be carried out to extend them to hypermedia [OHRE97]. This is again a place where the AHM is applicable, since it makes explicit the aspects of a hypermedia presentation that need to be expressed in a style sheet. Work on a suitable style sheet document model is in progress.

7.3.3 Runtime aspects of a presentation

Aspects of a hypermedia presentation which the AHM does not address tend to be related to runtime issues and in particular issues related to the behaviour of large continuous media items being transported over networks with fluctuating available bandwidths. A number of aspects are already addressed in passing in the model, such as providing alternate data types for transmission based on user preference or available network bandwidth. Others, such as guaranteeing streaming of high-bandwidth video, are not. These require a solution at the level of the network protocol, such as guaranteeing network bandwidth available to an application. Work is being carried out on developing appropriate protocols, such as RTP [RTP95], RSVP [RSVP96], RTSP [RTSP96].

7.4 In conclusion

Information exchange has been a human activity for tens of thousands of years. The technologies used for recording and presenting 'documents' have developed from cave walls through papyrus and paper to computers. Computers make it possible to construct a complete processing environment for authoring, storage, play-back on different devices and document re-use. This is facilitated by a common underlying document model which separates document structure from presentation.

Our contribution has been to unify existing models of hypertext and multimedia into a richer hypermedia model and to improve computer support for these documents by stating the requirements for an authoring system and demonstrating the feasibility of implementation.

The work reported in this thesis is, however, only a part of a complete environment for the creation, storage, manipulation, transmission and play-back of hypermedia documents. Additional requirements are a generalised approach to processing documents, an agreed upon presentation model and a methodological approach to creating sets of hypermedia documents. The challenge is to ensure that these technologies will be as durable as the cave paintings of our ancestors.