Using the Amsterdam Hypermedia Model for Abstracting Presentation Behavior

Lynda Hardman, Dick C.A. Bulterman

CWI

We give a short description of the Amsterdam Hypermedia Model followed by examples of its use in a number of existing and planned applications. The main application to date has been as a basis of the multimedia authoring system, CMIFed, along with its ability to specify trade-offs for resource use. We discuss the model's potential for generating differing document formats, followed by future work on using it as a goal format for generating multimedia documents.

1 Introduction

We have developed the Amsterdam Hypermedia Model (AHM) to describe sufficient complexity of a multimedia presentation and its interactions with the user so that the essence of the presentation can be preserved from one platform to another. This includes specifications of the media items (atomic pieces of multimedia data) used, the temporal relations among the items, layout of the items and possible user interaction. The model was developed to provide a balance between expressiveness of the information being modelled and simplicity of application. In the extreme case a hypermedia presentation can be programmed directly in a non-specialist programming language, giving flexibility but allowing for only minimal reuse. A simple model, supported by easy-to-use tools, is in turn too restrictive to allow the creation of anything more than, say, the sequential presentation of a slide show. The creation of a useful model is to find a pragmatic trade-off between these two extremes.

In this paper we give a brief description of the model, then describe how we already use the model for different aspects of multimedia and hypermedia manipulation and presentation, in particular as the basis of an authoring system and for making trade-offs in resource usage. We describe current work on using the model to derive different format for describing a presentation, and lastly future work on using it as the basis for document generation.

2 The Amsterdam Hypermedia Model

The AHM is an extension to the Dexter hypertext reference model [7] adding timing constraints and link contexts to the basic hypertext[1] notions. The expressiveness of the AHM allows two extreme cases to be modelled: continually playing passive multimedia presentations without links; and semantically typed node/link structures without media items[2]. The former requires specification of the data elements included in the presentation (video, audio, etc.) and their spatial and temporal relations. The latter has an emphasis on link and node types with perhaps only passing reference to data items (if any) related to the structure. A combination of these two model extremes is a rich information specification allowing both explicit temporal and spatial presentation relations closely integrated with typed structural specifications.

The model has been described elsewhere [10], but for completeness we give a brief overview here. We clas-

[1] Note that the Dexter model is by no means exclusively text based, but incorporation of time-based media is done only at a structural level and omits explicit timing relations within the model.

[2] The latter is again similar to the Dexter model, where the main difference is that explicit temporal relations can be expressed.

sify the elements of the model into structural and presentational elements, where the latter includes both spatial and temporal layout. Given the importance of temporal constraints in multimedia we discuss these separately.

2.1 Structural relations

The structure of a hypermedia document is built up of components connected by links via anchors. A component can be an atomic component, link component or a composite component. An *atomic component* describes information relevant to a single media item. A *composite component* is an object representing a collection of any other components

Anchors were introduced in the Dexter model [7] as a means of referring to part of a media item in a presentation in a media-independent way: they are an abstraction which allows data-dependencies to be kept hidden. The main use for anchors is to provide a source or destination object for linking among presentations, when they are used in conjunction with links. Another use is to provide a base on which to attach temporal relations so that internal points within media items can be synchronized with one another. Anchors can be defined in text as, for example, text strings and in images as a part of the image. Anchors in audio and video are conceptually similar, but technically require slightly more complex specifications [3], [5].

Composition can be either time dependent or time independent. *Time-dependent composition* allows the grouping of two or more nodes into one composite node along with their corresponding timing relations. Examples are parallel composition where items start together, and serial composition where one item starts when another finishes. *Time-independent composition* allows the grouping of items that have no time relations with each other. They *may* be played at the same time, because of a user following a link for example, but there is no pre-defined relation between them.

Links are defined as part of the Dexter model for explicitly representing relations among objects. They specify a logical connection between two (or more) end points specified via anchors. Most hypertext systems allow a user to follow a link as the basic form of interaction with the document structure. The use of links in hypermedia similarly allows the user to make choices as to which presentations to view and captures this in the document structure. The problem with links in multimedia is that a presentation normally consists of a number of media items playing simultaneously, and any one of these may have its own duration. In other words, links are not from static text or image items but from a complete multimedia presentation. This leads to the question of where links fit into this more dynamic and complex document structure. The question is how many of the items are associated with each end of the link. For example following a link may result in the complete window being cleared and the new presentation being displayed. On the other hand, the scope of the information associated with the link may only be a part of the original presentation. We make this scope specification explicit and call it the *context* of a link [9].

2.2 Spatial relations

Spatial relations among objects in a presentation can be defined with respect to a window, or relative to another item (or group of items). In the AHM we explicitly define higher-level presentation objects called channels. *Channels* define areas relative to a window into which an object can be played, so that when window size is changed, either within the one environment or across several environments, the channels change in proportion. This means that a presentation is not defined for a fixed window size. Other properties can be associated with the channel, such as high-level presentation specifications. These may be media-independent, for example background color, or media dependent, for example font style and size. This is useful for making global layout changes to the presentation. (This high-level presentation specification is used as a default, and can be overridden for individual nodes.)

2.3 Temporal relations

Timing relations in the AHM can be defined between atomic components, composite components or between an atomic component and a composite component. This allows the timing of a presentation to be stored within the document structure itself and not as some unrelated data structure (such as a separate timeline). These timing relations are specified in the model as *synchronization arcs*. These can be used to give exact timing relations, but can also be used to specify tolerance and precision properties which are needed when interpreting the desired temporal relation in a real-time environment. The end of a synchronization arc may be a component, but may also refer to a (single) anchor within a component, allowing constraints to be specified between internal parts of media items.

3 Using the AHM

We discuss here a number of past, current and future applications of AHM. Where these are reported elsewhere we give a brief description along with some references.

3.1 Basis of the authoring system CMIFed

We have used the AHM as a basis for our authoring system CMIFed[3] [15]. Our approach has been to take the model of a hypermedia document and to build an interface which supports both an author's approach to working with the narrative as well as providing direct access to the document structure [8]. A discussion of a range of authoring approaches for multimedia is given in [11]. Two of these approaches are to (a) program the presentation in terms of what happens next on the screen and (b) state the timing and layout relations among items declaratively and leave it to an interpreter to derive the actions required. It is this latter approach we take with CMIFed. Here, the author is protected from having to produce tedious procedural specifications (for example, place this picture on the screen in area A, then play this subtitle in area B), and can concentrate on creating relations among the objects (such as this subtitle goes underneath this picture). This allows greater flexibility in changing both small and large parts of the presentation.

Large structural changes can be made easily to the document by making the underlying document structure explicit and editable. Within CMIFed the structure can be time-dependent (with a choice of parallel or serial synchronization) and time-independent. In the former case, the main timing relations of the presentation are derived from the structure. The latter case allows separate presentations to be played in parallel.

3.2 Mapping media resources

The channels used in CMIFed are not only a means of defining window real estate for displaying media items, but specify separate information streams. This allows the creation of documents with multiple streams, such as multiple language subtitles and audio streams, where the end-user is given the choice of which stream to read/listen to. The basic document structure remains the same, and multiple streams are defined as part of a composite component.

Another use for multiple channels is to define multiple data formats for the same information, and allow the user to select the most appropriate for their environment. For example, in a presentation about the paintings of Mondrian, for each painting there is a high-resolution color representation, a low-resolution color and a black and white image. Depending on the end-user's machine capabilities and the availability of network bandwidth, the user can choose which representation to have transmitted across the network. Future work in this direction is to look at more automatic selection of the media channels depending on fluctuating available network bandwidth [4].

The channel objects possess a number of attributes. These allow global changes of presentation style, for example, changing the font used in all the headings (played in the same channel). While this is useful for the author for making global changes throughout the presentation, it is also useful for mapping the channels to the environment the presentation is being played on. For example, if a particular font has been specified for a number of text channels that is unavailable, the system can specify an available font to be used in its place. The text objects themselves are not changed, but are played back through the channels with their new font attribute.

3.3 Generating different document formats

By storing the elements of a hypermedia document in an expressive and high-level format, which we use for editing, this can be used to generate other more final-form oriented formats. Current work is investigating converting the CMIF document structure (that used by CMIFed) to HyTime, [6], [13]. Similar methods could be used for converting to other formats such as MHEG [14], from either the CMIF representation or the derived HyTime representation.

While we would be interested in investigating the conversion of CMIF for playback on the WWW, [16], the current HTML specification is insufficient for representing fully-fledged hypermedia documents (temporal relations cannot be represented). Also, SGML-only representations do not take spatial relations into account directly—these are defined via (non-SGML) style definitions which specify the layout properties for a particular document structure.

[3] C*MIFed* = *C*WI *M*ultimedia *I*nterchange *F*ormat *ed*itor.

3.4 Automatic document generation

While CMIFed has proved its worth as an editor for creating hypermedia presentations, it still requires the author to state (almost) everything about a presentation explicitly. One of our goals is to reduce the time the author has to spend on specifying low-level details even further by, for example, adding automatic generation techniques to the environment. We are starting work [12] on extending a presentation generator, such as WIP [1], [2], to include dynamic media and to generate the temporal and spatial constraints needed for specifying a multimedia presentation and to combine this with a front-end such as CMIFed. A model such as the AHM is the base needed for communicating which information can be generated and which needs to be added manually.

Conclusions

The existence of a model such as the AHM allows the expressive power of various authoring systems and document formats to be compared. The strength of the AHM is that it states the types of information that requires to be specified for a hypermedia document and places these within a framework— it does not force an author or system developer to specify all these types of information. If, however, a system is to support them then the model specifies how these are related to each other. In the case of CMIFed, the parallel development of the model and the authoring system proved invaluable, allowing conceptual additions to the model to be checked for their utility in the authoring process.

The model is not, as is the case for models in the field of knowledge representation, domain specific, but is form specific. It addresses the issues of combining elements occupying time and/or space (video clips, sound fragments, text and images) into a coherent multimedia presentation. It does not address any of the semantic issues of the contents of the presentation.

While the emphasis of the model is on combining data types such as video, audio, text and images, other data types are by no means excluded. So long as the temporal and space resource use for an element can be specified[4] then it can be combined as part of a hypermedia presentation. Simulation models for example can be incorporated, and in fact any arbitrary piece of code. HTML channels are not only possible in principle, but are already implemented in CMIFed.

Acknowledgments

Guido van Rossum, Jack Jansen and Sjoerd Mullender designed and implemented CMIFed. Jacco van Ossenbruggen contributed significantly to the HyTime work.

References

[1] André, Elisabeth, & Thomas Rist, “The Design of Illustrated Documents as a Planning Task” in “Intelligent Multimedia Interfaces”, ed. Mark T Maybury, AAAI Press/MIT Press, 1993 ISBN 0-262-63150-4, pp 94 - 116.

[2] André, Elisabeth, & Thomas Rist, “Multimedia Presentations: The Support of Passive and Active Viewing”. Deutsches Forschungszentrum fur Kunstliche Intelligenz, Saarbrucken, Germany, Research Report RR-94-01.

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

[4] Bulterman, D.C.A., “Specification and Support of Adaptable Network Multimedia”, ACM/Springer *Multimedia Systems*, 1(2) September 1993. `http://www.cwi.nl/ftp/mmpapers/mm.systems.ps.gz`

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

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

[4] In the case of a continuous video stream, for example a live feed, then the space constraint is known and the time usage is unlimited. The user, however, does not have to wait until completion before jumping to somewhere else in the presentation.

[7] Halasz, F. & Schwartz, M., “The Dexter Hypertext Reference Model”, *Communications of the ACM* 37(2), Feb 94, pp 30-39.

[8] Hardman, L., Van Rossum, G. & Bulterman, D.C.A., “Structured Multimedia Authoring”, *Proceedings of ACM Multimedia ’93*, Anaheim, Aug 93, 283 - 289. `http://www.cwi.nl/ftp/mmpapers/sma.ps.gz`

[9] Hardman, L., Bulterman, D.C.A. & Van Rossum, G., “Links in Hypermedia: the Requirement for Context, *Proceedings of Hypertext 93*, Seattle, Nov. 93, pp 183-191. `http://www.cwi.nl/ftp/mmpapers/ht93.ps.gz`

[10] Hardman, L., Bulterman, D.C.A. & Van Rossum, G., “The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model”, *Communications of the ACM* 37 (2), Feb 94, pp 50-62.

[11] Hardman, L. & Bulterman, D.C.A., “Authoring Support for Durable Interactive Multimedia Presentations”, STAR Report in Eurographics ’95, Maastricht, The Netherlands. `http://www.cwi.nl/ftp/mmpapers/eg95.ps.gz`

[12] Hardman, L. & Bulterman, D.C.A., “Towards the Generation of Hypermedia Structure”, Proc. of First International Workshop on Intelligence and Multimodality in Multimedia Interfaces, Edinburgh, UK, July 1995. `http://www.cwi.nl/ftp/mmpapers/immi1.ps.gz`

[13] HyTime. Hypermedia/Time-based structuring language. ISO/IEC 10744:1992.

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

[15] G. van Rossum, J. Jansen, K S. Mullender and D. C. A. Bulterman (1993). CMIFed: a presentation environment for portable hypermedia documents. In Proceedings: *ACM Multimedia ’93*, Anaheim CA, Aug, 183 - 188. `http://www.cwi.nl/ftp/mmpapers/cmifed.ps.gz`

[16] Schatz, B. R., & Hardin, J. B. (1994). NCSA Mosaic and the World Wide Web: Global Hypermedia Protocols for the Internet. *Science 265*, 12 August 1994, 895 - 901.