System design for structured hypermedia generation

Marcel Worring Carel van den Berg
Computer Science and Logic, University of Amsterdam
Kruislaan 403, 1098 SJ, Amsterdam, The Netherlands
worring@fwi.uva.nl

Lynda Hardman
Multimedia Kernel Systems, CWI, Amsterdam, The Netherlands

Abstract
In this contribution we consider the design of a hypermedia information system which not only includes standard functionality of storage and presentation, but also the automatic generation of hypermedia presentations on the basis of a domain dependent knowledge base. We identify and describe the data sources required and the processes involved.

1 Introduction

The usefulness of multimedia information systems hinges on the ease with which the information can be retrieved and on the speed and quality of the presentation of the information to the user. The most convenient way of interacting with multimedia information is through a hypermedia interface, where the user is guided in navigating through the large set of media items. This requires the definition of links relating the different pieces of information. This is a well known concept in hypermedia systems, but in many such systems the links are embedded within the media. Recent research on hypertext models [1], hypermedia models [2] and open systems such as MICROCOSM [3] have introduced the concept of link databases in which links are stored separately from the media, using the notion of anchors. In such open environments it becomes feasible to integrate multimedia information systems and hypermedia interfaces.

With the large variety of user platforms and user requirements it is virtually impossible for an information provider to anticipate the full set of hypermedia presentations one is likely to encounter. Therefore, rather than trying to generate all possible hypermedia presentations beforehand we aim at a multimedia information system providing tools to generate presentations automatically when they are requested in a certain context by the user. To do this automatically requires explicit knowledge about the domain. Example of applications where such domain knowledge is present are e.g. medicine, weather, sports, and news. In all of these domains the domain knowledge is fixed whereas the media items are changing constantly.

Providing high quality presentations to the user poses high demands both on the database storing the information, as well as on the presentation environment. We are currently designing a system containing the above mentioned functionality based on the extensible database system Monet [4] and the CMIF presentation environment [5].

In this paper we will consider some topics in the design of the proposed system. We will illustrate most concepts using video as it is the most complex and data-intensive media

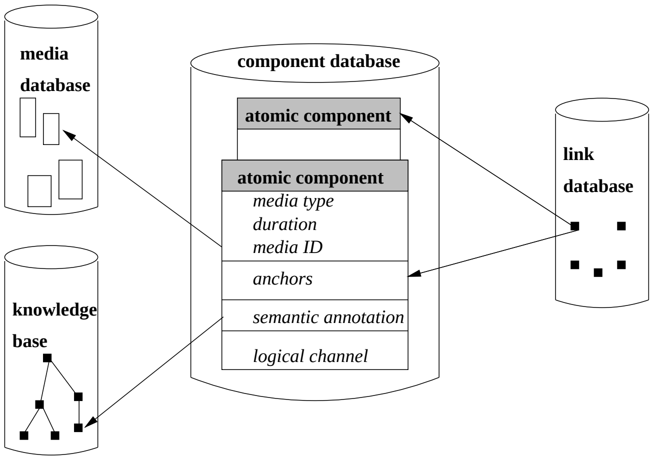

Fig. 1. *The logical databases used in the system and their relations.*

type. In section 2 the datamodel is introduced. Section 3 describes the processing steps used in the system and finally in section 4, a design for the architecture of the proposed system is presented.

2 Datamodel and logical databases

Given the complex functionality of the system it is important to have a precise definitions for the terms used for different types of information acted upon in the steps of the process. As datamodel we use the Amsterdam Hypermedia Model (AHM) [2], which can be viewed as an extension of the Dexter model [1].

The *media items* are the raw pieces of data e.g. a piece of video, or sound. The *atomic components* in the system are objects having a reference to a media item as well as an identification of the *media type* and for dynamic data its *duration*. Objects embedded in the medium, such as an object in a picture, are identified using *anchors*. Each atomic component has a list of anchors defined within the scope of the component. Note that anchors are *not* encoded within the media items. A number of components can be grouped into a *composite component* with explicit timing relations among them. In the annotation phase each component will receive a *semantic annotation* which is an instantiation of one or more concepts in the knowledge base. *Links* connect to atomic or composite components, or to anchors within a component. They can also be specified as database queries. Each atomic component is assigned a *logical channel* as presentation specification, which is an abstraction of a *physical channel* capable of playing the associated media item.

The information on presentations in the above described format is stored in a number of logical databases: the media database, the knowledge base, the link database and the component database. To achieve independence in the processing steps presented later, these databases do not have symmetric relations. The exact relations are shown in figure 1.

3 Processing steps

Let us first consider an example of interaction with the proposed system. A person is consulting a database on animals and is looking at a multimedia presentation about the Southpole. At a certain point in a video on the animals living on the Southpole the user sees a penguin and decides that more information on penguins would be interesting and clicks with the mouse on the visual representation of the penguin in one of the frames. If a link has already been created by an author, the user can follow it, however, there might

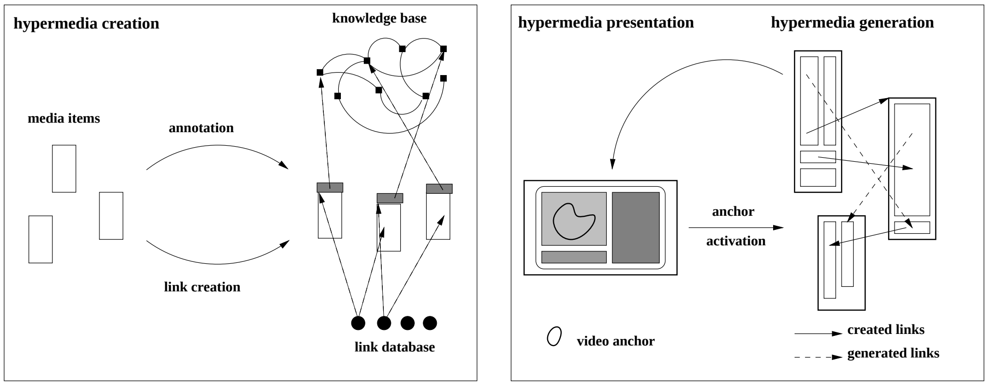

Fig. 2. *Overview of the different steps.*

be no link from the penguin to other components or anchors. Now, if the object in the video had an attribute stating that it were a penguin we could retrieve all media items from the information system which are in some way related to penguins. This requires that we have a knowledge base describing the domain. Retrieving this information gives us a collection of media items with possibly some links from the link database and virtual links derived from the knowledge base. This collection of media items are then structured automatically, combining them into coherent groups, e.g. all information on different species of penguins and one presentation on their diet. This results in a new hypermedia document which can be played at the user's hardware.

The example indicates some of the information processing carried out by the system. In general we can divide the processes into three main steps. In the *hypermedia creation* step basically all steps are done which are carried out prior to, or at the time of entering the media items into the system. To be precise, it involves the creation of the knowledge base, insertion of the media items and giving annotations, and finally the entering of links into the link database. The other two steps are performed at run-time and will be performed in an alternating sequence. These steps are the *hypermedia presentation* (playing) of hypermedia documents and the *hypermedia generation* performed whenever the user wants to follow links other than those foreseen by the author. The different processing steps are illustrated in figure 2 and will be described in the following sections.

3.1 Hypermedia creation

3.1.1 Knowledge representation

Domain knowledge needs to be represented in different ways for different purposes. Our interest is limited to the task of finding media items similar to another item. That is, we are concerned with the similarity of different pieces of information and are not concerned with interpreting what those pieces of information actually mean or represent. Porter notes that the types of features used for processing the semantics of items are of crucial importance [6]. In particular, superficial features which are about form and are independent of context and goal-of-use, e.g. size, color, and material, should be distinguished from abstract features which are about function and are dependent on the context and goal-of-use, e.g. "hammer" (a tool for hitting). Superficial features can be derived from the media items through data analysis. Abstract features are much harder to obtain but might often be more useful in matching. An example of a system based purely on superficial features is the QBIC system [7] whereas in e.g. CORE [8] the distinction between the two types of features is made explicit. Two ways of obtaining abstract features are by hand or via a domain knowledge representation. Our approach is to first create a domain knowledge representation by hand, and to assign aspects from this to the anchors within the media items by hand. This can be combined with an analysis of the raw data of the media items to obtain superficial features.

As concerns domain based annotation, work has been carried out by Davis [9] for the particular case of video. His task was slightly different to our own, but sufficiently similar to form a basis for this work. His chosen representation for domain knowledge of a collection of video sequences is based on knowledge frames, to be more specific the Framer system [10]. In broad terms this is a hierarchical frame-based structure, allowing multiple values for slots, where any node in the structure (leaf or interior) can be used for describing the persons and objects in the video as well as the activities performed by the subjects. For example a video showing a penguin walking on a ice-flow would have the description "penguin" as a semantic attribute, which is a specialization of "bird", which is a specialization of "warm-blooded creature", etc. Another attribute used would be "standing" which is a specialization of "pose" and so on. The representation in Davis is very broad. In [11] more narrow domain models are used, geared towards documentary video and news programs.

The knowledge base should also be capable of providing a similarity measure for two items. This is done by considering the hierarchical organization of the knowledge. Each semantic annotation consists of a number of attributes. The similarity between two attributes of different annotations can be defined as the number of steps one has to make in the hierarchy when moving from one concept to another. Hence, when given two annotations, a set of values is returned one for each attribute of the annotation. A zero value indicates that the match is exact and a positive value indicates that the match is in-exact. Going back to the previous example, consider a picture showing a penguin lying on a beach. Both the video and the picture would have the semantic attribute "penguin" so this attribute matches exactly. The semantic attributes "lying" and "standing" have a distance of 2, one from "lying" to "pose" and one from "pose" to "standing".

Given a set of annotations corresponding to components or anchors we can define a similarity matrix or graph giving the similarity measures among all items in the set. Such a similarity matrix will be used in the hypermedia generation step. In the next section we will first discuss the annotation step.

3.1.2 Automated video annotation

In both [9] and [11] the basic items to be considered for annotation are the segments (set of subsequent frames defining for example one shot) of the video. However, as concerns the knowledge representation there is no need to use segments only, the annotation could also be done on individual objects presented within the video, further called *video objects*. This gives a more direct association of the objects with their semantics. It should furthermore be noted that the information in the knowledge base is not restricted to video, but can also be applied to related media items like textual descriptions or audio fragments.

Annotating a video in the above manner with the aim of defining links connecting to a specific shot or a video object, requires identification of the shot boundaries and the objects appearing within the shots. Doing this in a fully interactive way is a tedious task as first shots have to be defined (probably using a hierarchical magnifier as described in [11]) and then objects have to be outlined in every subsequent frame. Using computer vision techniques the user can be aided in these two tasks.

Methods for detecting shot boundaries automatically are plentiful e.g. [11], [12], [13]. It can be concluded that accuracies of 90-95% can be achieved with the available methods. In an interactive environment this will in general be sufficient. In [14] the use of video objects as anchors is proposed. Their method of defining anchors aids the user in defining

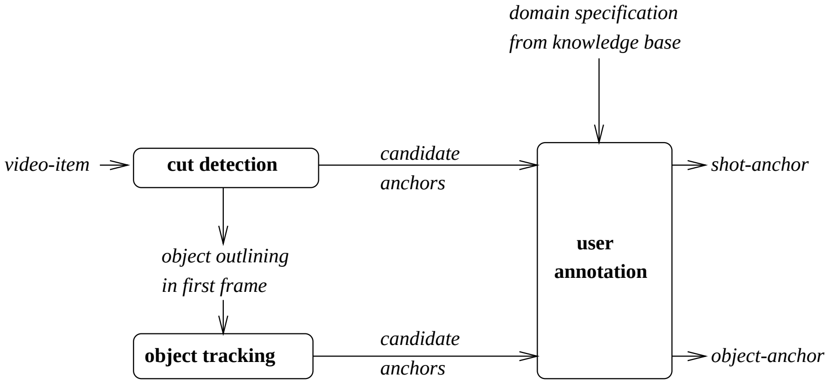

Fig. 3. *Overview of the annotation process.*

objects by using interpolation. However, this is purely based on computer graphics and no use of the video data itself is made. Hence, the resulting video objects have an inaccurate representation.

We intend to use an object tracking framework similar to [15]. In this framework a contour is parameterized with a small set of parameters using B-splines. Based on the video data found locally around the contour the object is tracked through the sequence using predictive filtering. For our purposes we will extend the framework to take into account color edges rather than intensity edges. This might be combined with the use of multi-layered video segmentation to find coherently moving objects [7]. In restricted domains the use of explicit models for detecting shot boundaries is feasible [11], but this will not be considered here. The automated video annotation process is illustrated in figure 3.

3.1.3 Link database creation

Although we aim at automatic generation of links the author can always create a link database. For links based on information not explicitly coded in the knowledge base this is even essential. Adding links which can be derived from the knowledge base should be done in close conjunction with the hypermedia generation step described in the next section. Adding those to the link database might speed up the processing, but adding them to the link database explicitly is not essential.

3.2 Hypermedia generation

Whenever the user selects a subject of interest, by selecting an anchor in a media item, for which more information should be provided, the semantic attributes associated with the selected anchor are recorded. A call is made to the database to select items that are similar to the given set of attributes [1]. This yields a set of media items as well as a similarity matrix describing their relations (see section 3.1.1).

The set of media items should be presented to the user in a structured way as a hypermedia presentation. Here we have to decide which items will be grouped into composite components and how they will be connected by links. This is done on the basis of a set of heuristics. Heuristics for semantic grouping can be based on the matching criteria in [16]:

- goal-directed: group components that involve the same goal;
- salient-feature: group components that match most important features or largest number of important features;

[1] In the ideal situation there would be some sort of memory, so that the user's previous selections can be stored and the associated attributes used to contribute to the information used in the database search.

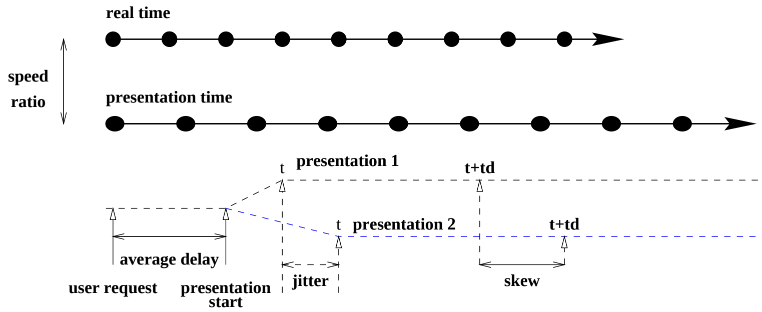

Fig. 4. *Quality of service.*

- specificity: group components that matches features exactly over those that match features generally;
- frequency preference: group components which are matched frequently;
- recency preference: group components which are matched recently;
- ease-of-adaptation: group components for which the features are easily adapted to new situations.

The above criteria can be mapped directly to the distances defined in the similarity matrix or their dynamic behavior. Example composition and linking heuristics not based on the semantics are:

- create the smallest number of composites;
- create the smallest number of links between composites
- don't allow incompatible media types (e.g. 2 videos together).

The weighting for the different criteria depends on how the different approaches work in practice, and may even be put in the hands of the end user e.g. to give preferences for an overview of a subject area, or an in-depth search.

3.2.1 Hypermedia presentation

In the above generation step a complete hypermedia presentation description is derived. At this point not yet including the actual media items, but a logical channel has been assigned to each media item. As indicated before a logical channel represents a physical channel capable of playing the media item. Hence when the hypermedia presentation is sent to the client logical channels have to be mapped to the physical channels available. When this mapping is performed the client informs the database server of the physical channel properties like e.g. resolution for a picture channel.

A key issue for acceptance of a multimedia information system is the quality of service provided by the system. The parameters which determine the quality of service are depicted in figure 4. Ideally real time equals the presentation time. The *ratio* between real time and the presentation time is an important quality factor. *Average delay*, *jitter* and *skew* represent relative timing delays between user requests and simultaneous presentations. The last factor, the *utilization* describes the ratio between the data volume used for the presentation and the data volume available.

From this description it is clear that the quality of service is timing related. Consequently, this aspect has a great impact on the system design as a whole. As the multimedia data used for a presentation is in general too voluminous to be stored in the client the database must be designed to offer the quality of service. In other words it must have a real-time kernel.

To keep a specific quality of service it might be required to reduce the utilization such that all of the above measures are kept within acceptable limits. This can for example be achieved by sending images at a reduced resolution. This adaptation can also be initiated

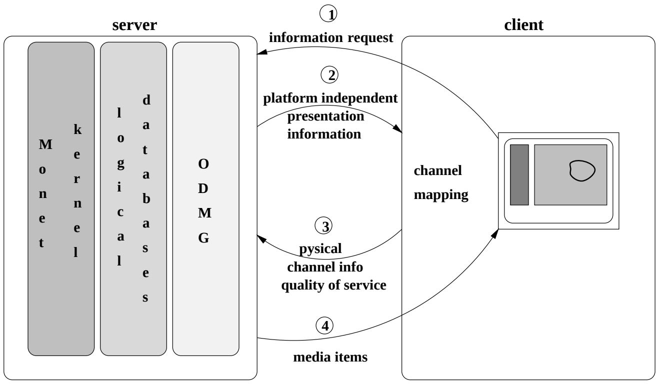

Fig. 5. *Overview of the layered architecture of the multi-media information system (or server) and the communication with the client in the presentation of hypermedia.*

by the the client when for example a window in which a media item is presented is resized. So apart from sending information on the properties of the physical channels also the required quality of service should be communicated. On the basis of these measures the database can start sending the actual media items to the client in the appropriate format and with the highest quality possible within the constraints. The whole process is shown in figure 5.

4 System architecture

The limitations for storing multimedia information in relational systems are well known. The kernels of these systems simply do not provide the hooks for achieving the fine control required in meeting the real time constraints. Object-oriented systems have the advantage that they enable the expression of multimedia data operations, such as image and video algebra [17]. Unfortunately, real-time behavior is not achievable with these systems.

Extensible databases provide the required level of control to implement multimedia databases. They allow extension of a small fixed database kernel with application specific data types and operations. Our system is based on Monet [4], an extensible main-memory database system. Monet uses a flexible and efficient decomposed storage model and offers database triggers, a type extension mechanism, and a set of binary relational algebra operations. The latter have predictable performance which is important in achieving a certain quality of service.

The basic media types are implemented using the type extension mechanism of Monet. These types include video, audio, images, links, anchors and the spatial and temporal relations used for presentation. At the application interface level an Object Oriented interface to these types is provided based on the ODMG datamodel. This has the advantage that a seamless interface is provided for applications written in any language. Currently a binding for C++ exists and a binding for Java is being considered.

5 Conclusion

We have considered the design of a multimedia information system in which hypermedia presentations can be generated automatically. In the system we have identified a number of logical databases namely a knowledge base, a media database, a component database, and a link database. The processes acting upon this information can be divided into three steps. A *hypermedia creation* step in which the logical database are populated. In this step each component is assigned a semantic annotation in semi-automatic way. The *hypermedia generation* is based on heuristics using a similarity function for grouping and linking components. Finally, the design of the hypermedia system architecture is such that real-time adaptive *hypermedia presentation* can be achieved.

References

[1] F. Halasz and M. Schwartz, "The Dexter hypertext reference model", *Communications of the ACM*, vol. 37, no. 2, pp. 30–39, 1994.

[2] L. Hardman, D.C.A. Bulterman, and G. v. Rossem, "The Amsterdam hypermedia model: Adding time and context to the Dexter model", *Communications of the ACM*, vol. 37, no. 2, pp. 50–62, 1994.

[3] H. Davis, W. Hall, I. Heath, G. Hill, and R. Wilkins, "MICROCOSM: An open hypermedia environment for information integration", in *proceedings of ECHT*, 1992.

[4] Peter Boncz and Martin.L. Kersten, "Monet: an impressionist sketch of an advanced database system", in *BIWITT'95*, 1995.

[5] G. van Rossum, J. Jansen, S. Mullender, and D. Bulterman, "CMIFed: a presentation environment for portable hypermedia documents", in *Multimedia*, 1993.

[6] Porter et al, "Concept learning and heuristic classification in weak-theory domains", *AI Journal*, 1990.

[7] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system", *IEEE Computer*, vol. 28, no. 9, 1995.

[8] J.K. Wu, A.D. Narasimhalu, B.M. Mehtre, C.P. Lam, and Y.J. Gao, "CORE: a content based retrieval system for multimedia information systems", *Multimedia systems*, , no. 3, pp. 25–41, 1995.

[9] M. Davis, *Media streams: representing video for retrieval and repurposing*, PhD thesis, MIT, 1995.

[10] K. B. Haase, "FRAMER: a persistent portable representation library", in *Proceedings 11th European Conference on Artifical Intelligence*, 1994.

[11] S.W. Smoliar and H. Zhang, "Content-based video indexing and retrieval", *IEEE Multimedia*, pp. 62–72, 1994.

[12] A. Hampapur, R. Jain, and T. Weymouth, "Digital video segmentation", in *Proceedings of the Second ACM International Conference on Multimedia, San Francisco*, 1994, pp. 357–364.

[13] K. Otsuji and Y. Tonomura, "Projection detecting filter for video cut detection", in *Proceedings of the First ACM International Conference on Multimedia*, 1993, pp. 251–257.

[14] V. Burrill, T. Kirste, and J. Weiss, "Time-varying sensitive regions in dynamic multimedia objects: a pragmatic approach to content based retrieval from video", *Information and Software Technology*, vol. 36, no. 4, pp. 213–223, 1994.

[15] A. Blake et. al., "A framework for spatiotemporal control in the tracking of visual contours", *International Journal of Computer Vision*, vol. 11, no. 2, pp. 127–145, 1993.

[16] J. Kolodner, "Judging which is the "best" case for a case-based reasoner", in *Case-based reasoning workshop*, 1989.

[17] Ron Weiss, Duda Duda Andrzej, and David K. Gifford, "Content-based access to algebraic video", in *International Conference on Multimedia Computing and Systems.* MIT, Programming Systems Research Group, May 1994, pp. 140–151.