Multimedia reporting: building multimedia presentations with query answers

Augusto Celentano and Ombretta Gaggi

Dipartimento di Informatica, Università Ca' Foscari Venezia
{auce,ogaggi}@dsi.unive.it

Abstract. A *multimedia report* is a multimedia presentation which integrates data returned by one or more queries to a multimedia database, thus extending the concept of report familiar in traditional structured databases. In such a scenario information retrieval consists in building a continuous presentation in which the retrieved data are located, connected, synchronized and coherently presented to a user.
We discuss modelling of multimedia reports in terms of data co-ordination and synchronization, based on a synchronization model we have defined for specifying complex multimedia presentations. As in a report the user can browse the returned data without loosing consistency, in a multimedia report moving along the presentation time requires appropriate synchronization to be guaranteed.

1 Introduction

Multimedia databases are similar to structured databases concerning basic operations: classifying, storing and retrieving data through algorithmic procedures and interactive interfaces. Nevertheless, the presence of different media types adds new facets to the operations and increases the complexity of data management.

In this paper we approach a quite traditional problem in the new perspective of multimedia data retrieval. The problem is basically the construction of a *report* on a set of data, and can be stated as follows: given a set of multimedia data, a retrieve task consists in building a continuous presentation in which the retrieved data are located, connected, synchronized and coherently presented to a user.

Reporting is one of three basic access modes to a data repository, the other two being browsing and querying. *Browsing* means to access a data repository along a priori undefined paths according to an estimate of relevance that the user formulates as he or she proceeds in the exploration of the repository. *Querying* means to identify relevant information according to precise and pre-defined selection criteria based on the content of the information.

Accessing data by reporting means that the retrieved data are meaningful as a collection, and the relationships among data items are perceived as relationships among aggregations which have a meaning in the application domain. Also the presentation schema of the report suggests the user a way of reading it and adds further semantics to the data.

When the operating environment evolves from text-only databases to complex and distributed multimedia repositories, reporting must be extended to face issues like media delivery and synchronization, channel management, user interaction, and cannot be effectively performed without a suitable model for describing the multimedia presentation which constitutes the report itself.

The problems of automating the construction of multimedia presentation have been approached by several authors, that we review in Section 2. We focus our discussion to data co-ordination and synchronization, based on a media synchronization model we have defined for specifying complex presentations made of continuous media.

We assume the World Wide Web as the surrounding environment of our discussion. As a first consequence, we are concerned with scenarios in which media items might be delivered independently, possibly by several servers, and must be synchronized at the user client site. Then, since data instances are not known in advance, we are concerned with a data integration and synchronization model which is based on data classes and types rather then on data instances, a quite

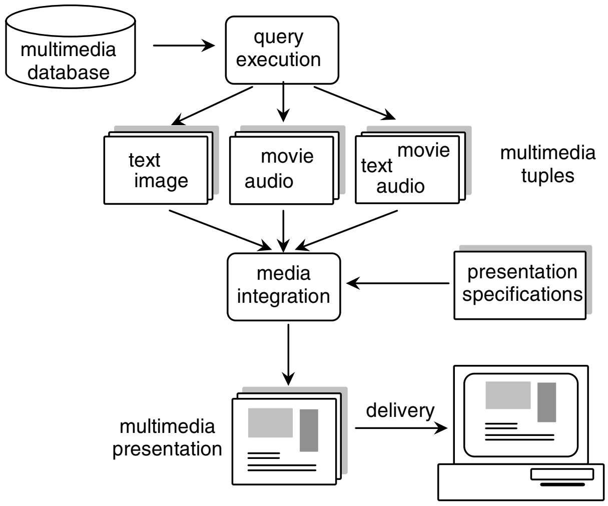

Fig. 1. Overview of the multimedia reporting process.

different situation with respect to traditional multimedia authoring and integration environments. Last, we are interested in building complex and interactive multimedia presentations. Therefore, we must define how different types of data can be presented together and how user actions must be handled in order to maintain the whole presentation coherent.

Figure 1 summarizes the process of building a multimedia presentation by querying a multimedia repository. Query execution returns a set of multimedia tuples whose components belong to different media types. In each tuple instances are homogeneous[1] therefore the specifications for media playback are compatible over all the data instances of a specific tuple type. The presentation specifications describe, among other information, how media items are integrated; the specifications describe how media items behave when events related to their playback happen. Master-slave relationships are established among media types, so that when the overall presentation is delivered to the user, its behavior is dictated by the dynamics and the timing of a master medium, usually a video clip or an audio track. Once the presentation is started, the master medium generates events when it starts, ends, or when the user interacts, e.g., by stopping playback. Such events propagate to other media, which in turn generate other events, determining the dynamic evolution of the whole presentation.

In this paper we are concerned only with the temporal and synchronization aspects of integration of the returned media items. We therefore do not face query specification and execution, nor we deal with layout specifications of the generated presentation. In Section 6 we shall discuss some open issues about multimedia querying.

The paper is organized as follows: in Section 2 we review the relevant literature by comparing the different goals and approaches taken. Section 3 introduces the functionality of multimedia reporting in terms of co-ordination and synchronization among the media objects returned. In Section 4 a suitable model for multimedia synchronization is presented, while Section 5 illustrates how such a model can be applied for specifying multimedia reports. Section 6 discusses consistency issues related to multimedia integration, and Section 7 presents concluding remarks.

2 Related work

Automation of the production of multimedia presentations has been approached in recent years from several points of view. All the approaches have the common goal of building in a more or less automated way a multimedia document that integrates different media objects extracted from a database or from document segments.

[1] In Section 6 we shall argue about this assumption.

SQL+D [1, 2] is an extension to SQL which defines the presentation properties of a multimedia query result. Given a multimedia database, an SQL+D query allows users to specify in the query a screen layout and time intervals to show the answer. In addition to SELECT-FROM clauses, DISPLAY-WITH clauses define screen areas (called *panels*), in which groups of retrieved media items are placed with specified relative positions. A SHOW clause defines the temporal behavior in terms of timed sequences of returned instances display.

The Cuypers system [15–17] is a prototype multimedia transformation environment supporting semi-automated assembling of multimedia documents according to a rich structural and semantic annotation based on XML. The annotation allows different processing steps concerning semantic structure, constraints satisfaction and final form presentation, which occur in multimedia authoring, to be integrated in a single execution stream.

In [10] the authors present a methodology for automated construction of multimedia presentations. Differently from the works reviewed above, the paper does not deal with query or retrieval environments, but focuses on the semantic coherency of a multimedia presentation in terms of inclusion and exclusion constraints. Given a set of multimedia document segments, the authors discuss how a user selection of some segments should be completed or modified by including additional segments based on semantic consistency of their content.

In [3] a system for the automatic generation, integration ad visualization of media streams is described. The authors view teaching and learning experience as a form of multimedia authoring, so the solution proposed are strongly oriented towards the educational domain, although this system can be applied to different fields. The paper describes a method to integrate different media streams, such as a video captured during a university lecture and its audio track, through the use of different levels of granularity, and of one particular solution to each level. A timeline "road map" of the lecture, marked with significant events, is proposed for the visualization of multiple streams.

Delaunay^{MM}[8] is a framework for querying and presenting multimedia data stored in distributed data repositories. Media objects returned by a query are inserted into a multimedia presentation. Delaunay^{MM} uses profiles to design user-defined layout of a document and ad hoc querying capabilities to search each type of media item. The authors do not address any formal model for the specification of temporal synchronization of different objects.

In most of the reviewed systems the authors assume (coherently with the examples provided) that the data items retrieved or selected for building the presentation are in some way predictable and homogeneous, i.e., they can be combined in temporal sequences or spatial layout with an *a priori* specification. This could be stated as an explicit requirement in order to be able to build presentations which are coherent for a user, and is almost true for static items like text and images. However, dynamic media like video and audio may need additional specifications of temporal behavior in terms of co-ordination and synchronization, that we shall discuss in this paper. The assembly of a multimedia dynamic presentation thus requires suitable models for specifying structure and temporal scenarios of hypermedia documents.

Amsterdam Hypermedia Model [11–13] was the first serious attempt to merge document structure with temporal relations. AHM distinguishes between atomic and composite components: the former are simple media elements, like a video file or a text page, and the latter are composition of different objects grouped together according to synchronization relationships. Media items are associated to channels which represent the devices needed for their playback.

SMIL[19], Synchronized Multimedia Integration Language, is a simple markup language defined as a W3C recommendation. It is an XML application to describe temporal behavior of a multimedia presentation using tags. Synchronization is achieved through tag `seq` to render two or more objects one after the other, and tag `par` to reproduce them in parallel. Using attributes it is also possible to play segments inside the time span of an object.

In [7] a formal framework for verifying temporal synchronization of a presentation is defined, with the goal of determining whether a multimedia presentation is synchronized or amenable to synchronization.

Madeus[14], an authoring and presentation tool for interactive multimedia documents, describes synchronization among different objects through the use of timing constrains based on particular events, like the beginning or the termination of a media file. Temporal information is represent through a direct acyclic graph, and layout is specified by spatial relations like *align* or *center*.

In [18] the authors discuss a system for authoring and formatting hypermedia documents, named HyperProp. It uses composition to represent spatial and temporal information. HyperProp provides the structural views to graphically browse and edit the logical structure of a document, the temporal view to represent objects along a timeline and the spatial view to formatting objects layout.

In previous works [4, 5, 9] we have discussed the problem of authoring and navigating hypermedia documents composed of continuous and non continuous media objects delivered separately in a Web-based environment. We have introduced a formal model which defines a static structure and synchronization relationships among media objects belonging to a same presentation. Temporal relationships between different media are described according to the event-based approach. In [6] we have applied the model to the identification of the scope of a query answer in a set of multimedia presentations. More details on the model are given in Section 4.

3 Multimedia reporting

Let us suppose a fashion Company wants to offer Web users the possibility of looking at a personalized catalog of the last collection in the shape of a multimedia presentation performing a virtual fashion show. A repository stores multimedia information about the fashion collections. The user can query the repository selecting the set of data which set up the virtual fashion show.

We do not enter into details about the visual richness of such a show, because it is out of the scope of this paper. We simply note that the result can be built on several types of data: for example, a set of pictures which portrait models wearing the selected dresses, or a set of movie clips which play parts of real fashion shows in which the selected dresses are shown. Text pages can describe the collection items from a stylistic point of view, and other text documents could provide details about tissues, techniques, commercial information, and so on. In order to complete and make more pleasant the presentation, a soundtrack could be played with songs or spoken comments.

Can we call such a presentation a "multimedia report"? Indeed, several aspects of text-based reports are present also here. The simplest multimedia presentation is a *slide show*: an ordered sequence of images displayed sequentially with appropriate timing and visual transitions, possibly with associated text documents. Such a presentation is similar to a report on an alphanumeric database because it exhibits a high degree of homogeneity in handling the components of the presentation: it is an organized collection of data, the overall structure reveals a recurring pattern, the transitions between data instances are perceivable like the breaks between groups in a report. Basically a slide show is a report translated from a linear space dimension to a linear time dimension.

Some of the systems reviewed in Section 2, e.g. SQL+D, can generate multimedia reports of such kind. Sequence, transitions, timing and spatial layout can be described by DISPLAY-WITH-SHOW clauses much as report sections are described by constructs of a reporting language. We can extend this simple scenario by introducing dynamic media which require complex synchronization specifications, e.g., by introducing movie clips and a soundtrack possibly made of different songs. In this case two data sets must be merged, and the transition between instances of the first set do not occur necessarily at the same time than in the second set, except for trivial cases. The slide show metaphor is no longer sufficient to describe such a scenario.

As an example, let us suppose that the presentation integrates a collection of dresses in the shape of movie clips coming from different fashion shows. Text pages describe the dress models, and for each fashion show a different song has to be played. A dress change (therefore a movie change) in the same fashion show, however, must not cause a song change.

This presentation is built on three different data types: the movie clips, the soundtracks, and the text documents associated to the movies. They can be returned by different queries as long as the instances of movies and texts are related by some foreign key correspondence[2].

[2] how such queries can be formulated is out of the scope of this paper

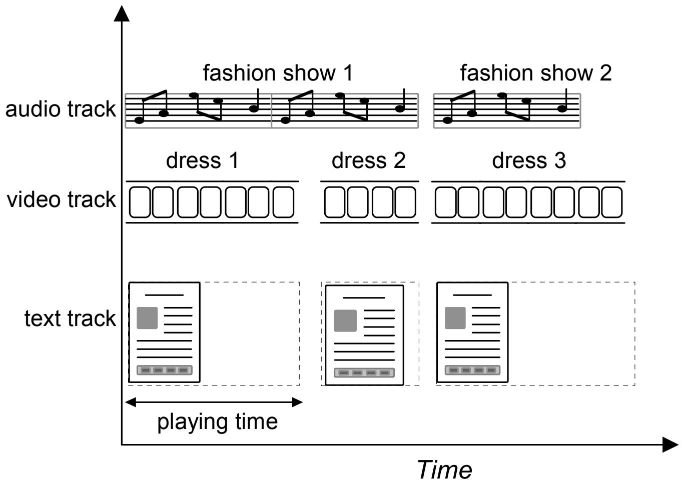

Fig. 2. Timeline of the example presentation.

We could attempt to describe such a presentation according to a timeline-based description like in Figure 2. Indeed, we cannot design such a timeline because we do not know how many objects will be returned by the query. Therefore we cannot define the transition times between instances. We do not even if text instances and movie instances are retrieved consistently, i.e., if for each movie there is a corresponding text page. Such a lack of information does not allow us to rely on models like the ones reviewed in Section 2. These models provide timing specification in terms of media sequencing and duration. In our example, moving from one fashion show to another causes not only a different movie to be played, but also a different soundtrack to be played. This event cannot be described in terms of media properties, but concerns the dynamic structure of the whole presentation

We need therefore to translate a representation based on a timeline into a *sequence of events* whose occurrence makes the presentation to evolve. In order to achieve this result we need a synchronization model that is able to define the relationships between the presentation of several media elements, defining also how the playback channels (e.g., the windows) are used and released.

Before describing this model we go someway further in order to justify the kind of synchronization relationships we shall introduce. The report composition model must allow us to go from a temporal description based on a timeline, like the one in Figure 2, to a description based on events like the one illustrated in Figure 3[3]. An event based description is more suitable for a Web environment since synchronization is not required to be fine-grained, but occurs only at specific events such as the beginning or the completion of a download, or the start and the end of a streaming medium, or a user action.

The model we have defined relies on relationships like "play medium B when medium A stops", which do not require to know in advance the duration and the physical features of the media involved, which are not known when the query is formulated and the presentation schema is defined.

4 Modeling synchronization in multimedia presentations

In previous works [4, 5, 9] we have proposed a model to describe synchronization among components of a multimedia presentation with reference to a distributed environment like the World Wide Web. The model defines relationships between media instances. It is however well adapt to describe relationships between classes, which in our scenario describe intensionally the query results.

Static components such as text and images are called generically *pages*; dynamic components are video and audio files. A hierarchical structure is defined which describes the overall presentation

[3] details of the figure will be described in next section

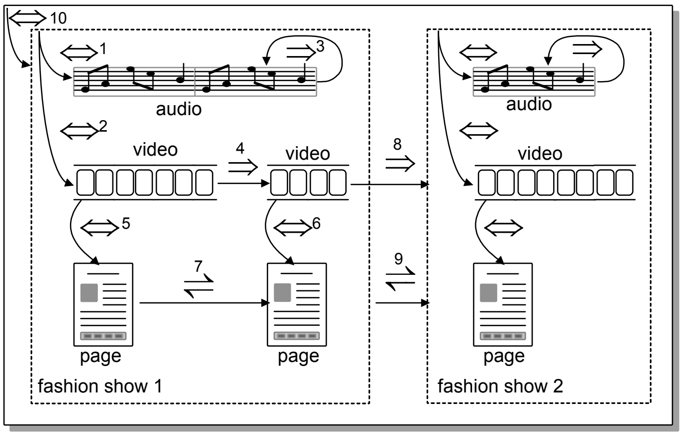

Fig. 3. Synchronization relationships between media items.

organization. The continuous media which constitute the main module content make up a *story*, which is composed of a sequence of continuous media files, audio and video, which are called *clips*. Clips are divided into *scenes*. A scene is associated to a (set of) static documents which are presented together. A clip with its scenes and associated pages build up a *section*.

Media objects require resources to be played. A *channel* is a virtual device allocated to a document media component, e.g. a window. Generally speaking, a channel is the set of resources a media object needs for its playback. Different media type require different channel types. A channel is *busy* if an active object is using it, otherwise it is *free* and can be used by another media object of the same type. Since static media do not evolve in time, they have an unlimited time extent, i.e., they holds a channel until forced to free it.

Synchronization is achieved with a set of primitives which define objects behavior during presentation playback and channels utilization. The events to which media react can be internal, like the beginning or termination of an object playback, or external, like a user action. We have defined five synchronization relationships.

- *A plays with B*, denoted by the symbol $\Leftrightarrow$ ($A \Leftrightarrow B$), to play two objects in parallel with object A acting as the master one; when it comes to end, object B play is also terminated.
- *A activates B*, denoted by the symbol $\Rightarrow$ ($A \Rightarrow B$), to play two objects in sequence; the end of object A causes object B to start playing.
- *A is terminated with B*, denoted by the symbol $\Downarrow$ ($A \Downarrow B$), to terminate two objects at the same time as a consequence of a user interaction or of the forced termination of object A.
- *A is replaced by B*, denoted by the symbol $\rightleftharpoons$ ($A \rightleftharpoons B$), to force the termination of object A so that object B can use the same channel.
- *A has priority over B with behavior α*, denoted by the symbol $\stackrel{\alpha}{>}$ ($A \stackrel{\alpha}{>} B$), to stop (if $\alpha = s$) or pause (if $\alpha = p$) object B when the user activates object A. This relationship describes the behavior of objects related by hyperlinks, that require to pause or stop part of a presentation in order to allow the user to focus the attention on the document at link destination.

Figure 3 shows an example of such relationships for the module described by the timeline of Figure 2. Relationships 1 and 2 state that the first audio track and the video clip showing the first dress start when the presentation of the first fashion show is started. By relation 3 the audio track is repeated as long as the presentation goes on. In this way we need not to be concerned with its length. Dresses are presented one after the other as described by relation 4 between the video clips. At the beginning of the first video clip, the first static page is presented, as described by relation 5, while the second page is displayed when the second video clips is played (relation 6). Since the two pages are displayed in the same window, the second one must replace the first one in the channel usage. This behavior is described by relation 7. The presentation of the first fashion show ends when the last video clip ends. Then, the second show is started, which takes the

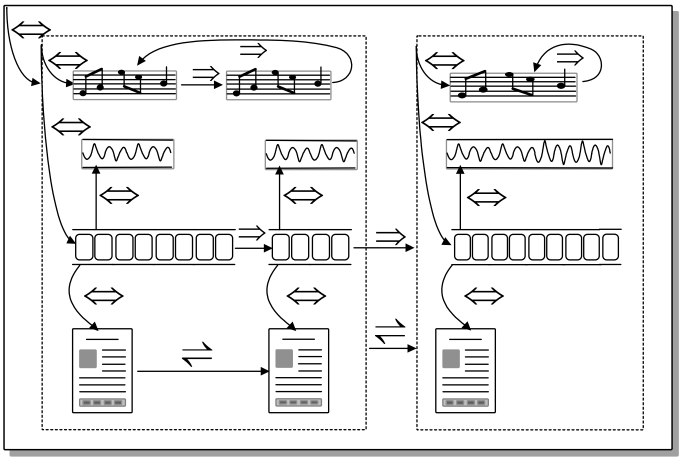

Fig. 4. A Synchronization schema for a more complex multimedia report.

resources that were allocated to the first part of the presentation. This is described by relations 8 and 9. The second fashion show proceeds in the same way, as well as the remaining ones. Relation 10 defines that the whole presentation starts with the first fashion show.

The whole presentation is structured in two levels, as revealed by the dotted regions they enclose media instances belonging to returned query results related to the same fashion show. The hierarchical structure allows the definition of environments in which some temporal relationships are derived by induction. For example, relations 1, 2 and 10. of Figure 3 are induced by the hierarchy between the whole presentation, its models (i.e., the different fashion shows), and the module components, therefore need not to be identified by the presentation designer, and introduce some degree of automation in multimedia authoring. The reader is referred to the cited previous works [4, 5, 9] for a more complete discussion of the model.

5 Presenting query results

The presentation of retrieved information can be described according to dynamic synchronization relationships, in order to present the user a coherent playback of the different media items involved. Let us modify and enrich our example, defining a multimedia presentation in which four kinds of media are involved, retrieved by queries from several related databases: movie clips, which portrait models with selected dresses; soundtracks, which play continuously in the background; text pages which describe some features about the portrayed dresses; and voice comments which introduce the different items of the fashion show, one short comment for each item. Also in this case the resulting presentation is organized along a hierarchical decomposition in two levels, the higher level defined by the aggregation of movie clips belonging to the same real fashion show (a group), the lower level defined by the sequence of movie clips portraying different dresses displayed in the same real fashion show. Figure 4 describes an instance of this presentation schema. In this case the voice comments are activated by the video clips, so that each clip starts a different comment. The soundtrack is still activated by the module which includes all the video clips of a specific fashion show, but executes several songs in sequence according to their own timing.

It is important to note that Figure 4 cannot describe *exactly* a presentation schema, since it has to be modelled intensionally and not extensionally, being unknown at presentation definition time the instances that will be returned by the query. It has the purpose of making evident what relationships need to be established among the returned media item in order to build a coherent view for the user.

The synchronization schema of this presentation is based on the following properties.

- The movie clips are the "master" section of the presentation. They give the overall timing to the presentation, because each of them defines the time a text page is displayed.

– The soundtrack is made of songs which are played one after the other within the same group. The time when a transition between two songs occurs is not related to other media, and depends only on the songs length.
– The voice comment is started by the movie clip to which it is associated.
– The whole presentation is a self-contained structure which begins execution by starting the execution of the first group, then letting each group start the following one upon completion. A group, in turn, starts the movie clip and the background soundtrack[4]. Starting from the presentation activation, every media item is played upon occurrence of events in the set retrieved by the queries, as described by the synchronization specification.

The constraints on media instances returned by the queries can be expressed by the following relationships. For each module:

$$\begin{array}{ll} soundtrack_{k,j} \Rightarrow soundtrack_{k,j} & \forall j \\ video_{k,i} \Rightarrow video_{k,i+1} & \forall i \\ video_{k,i} \Leftrightarrow voice_{k,i} & \forall i \\ video_{k,i} \Leftrightarrow textpage_{k,i} & \forall i \\ textpage_{k,i} \rightleftharpoons textpage_{k,i+1} & \forall i \\ module_k \Leftrightarrow video_{k,1} & \forall k \\ module_k \Leftrightarrow soundtrack_{k,1} & \forall k \\ module_k \rightleftharpoons module_{k+1} & \forall k \end{array}$$

where indexes i, j span over the media instances of module k. The first module is activated by starting the presentation and subsequent modules are activated in sequence by the end of the last video clip of the previous module:

$$\begin{array}{l} presentation \Leftrightarrow module_1 \\ video_{k,last} \Rightarrow module_{k+1} \ \forall k \end{array}$$

More care should be put in order to consider the possibility that the voice comments last longer than the movie clips in some cases. In general we cannot know about returned media instances all the properties that would be needed to anticipate their behavior. The relationships holding between modules, video clips and soundtracks (that we have defined *constraints* because are not known until query execution) should in this case specify that the medium which has the longer time span act as a master in activating the other medium:

$$\begin{array}{ll} max(video_{k,i}, voice_{k,i}) \Leftrightarrow min(video_{k,i}, voice_{k,i}) & \forall i \\ max(video_{k,i}, voice_{k,i}) \Rightarrow max(video_{k,i+1}, voice_{k,i+1}) & \forall i \\ module_k \Leftrightarrow max(video_{k,1}, voice_{k,1}) & \end{array}$$

where $max(a, b)$ and $min(a, b)$ return the longer and the shorter medium, in terms of time span. They are defined by functions at presentation specification time, and are expanded to specific instances identifiers when actual data are returned.

6 Discussion

A number of issues deserve discussion, due to the many unknown elements with respect to a fully defined multimedia presentation.

Inter-media consistency. We assume that query execution returns sets of results which are consistent by definition. This is a quite obvious assumption, inherited from the traditional database environment where tuples are homogeneous, but in principle it can be false. For example, some dresses could be described only by text pages without movies, or conversely only by movies without related texts. In order to manage these cases we could follow two ways: to extend the *master-slave* relationship between media types to identify relevant instances, not only timing relationhips, e.g.

[4] In the model terminology such groups are called *modules*

by assuming that the movie clip is the master data item, and a text without an associated movie clip is meaningless; or to supply stubs, i.e., empty placeholders for instances which do not have a counterpart in other media types, thus introducing some form of NULL values in multimedia data.

In both cases this issue can be approached by some kind of query post-processing, which should return a consistent and complete set of data, possibly with explicit NULL values. Both solutions are straightforward to implement, and the choice should be related to the desired meaning and appearance of the whole target presentation rather than to abstract considerations.

Boundary constraints. Linking distinct objects into a seamless multimedia composition requires compatible interfaces between components. Differently from textual database reports we cannot guarantee that returned data items can always be integrated in a multimedia presentation by showing the same visual properties. A complete discussion goes beyond the scope of this paper but we can address some main issues here. Given two generic data items, their compatibility is referred to a notion of type equivalence which is related to the set of values they can hold, the operation defined over them, the selection properties that allow to select parts of aggregations, and so on. Multimedia data have a much richer set of properties to be considered. For example, two movies are compatible or not according to a wide spectrum of features: size, color, resolution, frame per second, compression, and so on. Some of these properties are fixed, other can be modified without changing the data meaning. The presence of such differences makes compatibility a matter of substance rather than a matter of pure form.

As an example, the transition between two images is perceived differently not only according the visual properties, but also according to the meaning that the images convey. In a fashion show a sequence of images portraying models with different dresses in the same show room is perceived differently from a sequence of images taken in different rooms. We argue that this problem could be approached as integrity constraints are approached in conventional databases, in order to guarantee formal consistency of sets of related data. To some extent boundary constraints should be defined as a means to measure the consistency of the user perception of the multimedia report. However we feel that a solution to this problem is not close.

Query formulation and integration. We have not discussed issues related to the formulation of queries and to the syntactic and semantic devices needed to relate instances returned from different media repositories, e.g., movie clips, associated voice comments and related pages. This of course is a problem of crucial importance, and we do not claim it is easy to formulate formally and to solve. Automatic correspondence between different types of data cannot be established safely relying only on the interpretation of the data content. Content-based image retrieval systems available today are still far from handling a concept of similarity based on the human perceived meaning of the images. Extending the semantic interpretation to several media adds an unknown amount of complexity.

This problem can be approached with success only if we assume that multimedia reporting shares with textual reporting a correspondence between the syntactic and the semantic level of the query execution. In other words, multimedia data can be integrated by automatic procedures only if some features (tags, metadata, identifiers) can be fully recognized at a syntactic level, and used as semantic indexes to relate instances belonging to different data types. XML is a promising environment for approaching this issues, as its use in some related work demonstrates [15, 16].

7 Conclusion

Multimedia reporting can be viewed as an information retrieval activity which constructs a multimedia presentation integrating the data returned by queries directed to different repositories. The consistency of the presentation requires modelling of the relationships among the media objects which are returned by query execution. The relationships define the compatible execution of media elements in terms of synchronization relationships which drive the overall dynamics of the presentation.

In a previous work we have defined a model for delivering and synchronizing complex multimedia presentations. This model is suited to describe the relationships that must hold among elements of a multimedia report built automatically from results of multimedia database queries. While the spatial and logical composition of the media items returned into a multimedia document has been approached in the literature, the management of the synchronization relationships among the different components has not received too much attention. Our proposal makes a step in this direction, even if further investigation is needed.

References

1. C. Baral, G. Gonzalez, A. Nandigam. SQL+D: extended display capabilities for¡multimedia database queries. *ACM Multimedia Conference*. Bristol, UK, 1998.
2. C. Baral, G. Gonzalez, T. Son. A multimedia display extension to SQL: language design and architecture. *University of Texas El Paso, El Paso*, 1997.
3. J.A. Brotherton, J.R. Bhalodia, G.D. Abowd. Automated Capture, Integration, and Visualization of Multiple Media Streams. *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 54–63, Austin, Texas, June 28–July 1, 1998.
4. A. Celentano, O. Gaggi. A Synchronization Model for Hypermedia Document Navigation. *Proceedings of the 2000 ACM Symposium on Applied Computing*, pages 585–591, Como, 2000.
5. A. Celentano, O. Gaggi. Authoring and Navigating Hypermedia Documents on the WWW. *ICME 2001, IEEE International Conference on Multimedia and Expo*, Tokio, August 22-25, 2001.
6. A. Celentano, O. Gaggi. Querying and Browsing Multimedia Presentations. In M. Tucci (ed.), *MDIC 2001, 2nd International Workshop on Multimedia Databases and Image Communication*, LNCS Series 2184, Springer Verlag, 2001.
7. I:F: Cruz, P.S. Mahalley. Temporal Synchronization in Multimedia Presentations. *IEEE International Conference Multimedia Computer Systems (ICMCS '99)*, Vol. 2, pages 851–856, 1999.
8. I.F. Cruz, W.T. Lucas. A Visual Approach to Multimedia Querying and Presentation. *Proceedings of the Fifth ACM International Conference on Multimedia '97*, pages 109–120, Seattle, WA, USA November 9-13, 1997.
9. O. Gaggi, A. Celentano. Modeling Synchronized Hypermedia Documents. *Technical Report n. 1/2001*, Department of Computer Science, Università Ca' Foscari Venezia, Italy, January 2001, submitted for publication.
10. V. Hakkoymaz, J. Kraft, G. Ozsoyoglu. Constraint-based automation of multimedia presentation assembly. *Multimedia Systems*, 7, pages 500–518, 1999.
11. L. Hardman, D.C.A. Bulterman and G. van Rossum. The Amsterdam Hypermedia Model: adding time and context to the Dexter Model. *Comm. of the ACM*, 37(2), pages 50–62, 1994.
12. L. Hardman. Using the Amsterdam Hypermedia Model for Abstracting Presentation Behavior. *Electronic Proceedings of the ACM Workshop on Effective Abstractions in Multimedia*. San Francisco, CA, 4 November 1995.
13. L. Hardman. Modelling and Authoring Hypermedia Documents. *PhD. Thesis*, University of Amsterdam, 1998. `http://www.cwi.nl/ftp/~lynda/thesis`.
14. M. Jourdan, N. Layaïda, C. Roisin, L. Sabry-Ismaïl, L. Tardif. Madeus, an Authoring Environment for Interactive Multimedia Documents. *ACM Multimedia Conference*. Bristol, UK, 1998.
15. J. van Ossenbruggen, F. Cornelissen ,J. Geurts, L. Rutledge, L. Hardman. Cuypers: a semi-automatic hypermedia generation system. CWI Tech. Report INS-R0025, CWI, Amsterdam, 2000.
16. J. van Ossenbruggen, J. Geurts, F. Cornelissen, L. Rutledge, L. Hardman. Towards Second and Third Generation Web-based Multimedia. *Tenth International World Wide Web Conference*. Hong Kong, May 1–5, 2001.
17. L. Rutledge, B. Bailey, J. van Ossenbruggen, L. Hardman, J. Geurts. Generating Presentation Constraints from Rethorical Structure. *Proc. 11th ACM Conference on Hypertext and Hypermedia*. San Antonio, Texas, USA, May 30–June 3, 2000.
18. L. F. G. Soares, R. F. Rodrigues, D. C. Muchaluat Saade. Modelling, authoring and formatting hypermedia documents in the HyperProp system. *Multimedia Systems*, 8(2), 2000.
19. Synchronized Multimedia Working Group of W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. *W3C Recommendation*, 15 June 1998. `http://www.w3.org/TR/REC-smil`.