A CONSTRAINT-BASED APPROACH FOR THE AUTHORING OF MULTI-TOPIC MULTIMEDIA PRESENTATIONS

Elisa Bertino[1], *Elena Ferrari*[2], *Andrea Perego*[3], *and Diego Santi*[3]

[1] CERIAS, Purdue University, IN, USA
bertino@cerias.purdue.edu

[2] DSCPI, Università degli Studi dell'Insubria, Como, Italy
elena.ferrari@uninsubria.it

[3] DICo, Università degli Studi di Milano, Italy
{perego,santi}@dico.unimi.it

ABSTRACT

Synchronized multimedia applications play an important role in a Digital Library environment, since they allow one to efficiently disseminate knowledge among differently skilled users through an approach which is more direct than the classic 'static' documents.

In this paper, we propose a new authoring approach based on an innovative presentation structure and a new class of content-based constraints. Thanks to a flexible heuristic process, such features allow the author to easily combine several multimedia objects into a multi-topic presentation, whose different contents can be freely chosen by end users according to their preferences or skills.

1. INTRODUCTION

Many advanced multimedia applications—in particular, Distance Learning, Virtual Museums, and so on—can take great advantage by the use of synchronized multimedia—i.e., multimedia presentations which combine heterogeneous media objects played according to a given temporal sequence. Such applications play an important role in a Digital Library (DL) environment, since they allow one to efficiently disseminate knowledge among differently skilled users through an approach which is more direct than the classic 'static' documents. They are clearly more complex to model and to deal with, so it is important to define well-designed metaphors and models both for their specification and for the final representation. Although a wide variety of models have been proposed in the last five years for the specification of multimedia presentations, all the various solutions can be classified into two main approaches: the *operational* and the *constraint-based* approach.

The operational approach (see, e.g., [1–3]) makes use of system-defined structures to model both the spatial location of the media objects inside the visualization device (usually through (x, y) coordinates) and their temporal behavior, using timelines, trees, graphs, Petri-nets, or *ad hoc* scripting languages, which require the author to be aware of the particular operational structure used by each system. In general, in such an approach there is no particular distinction between presentation specification and representation, since the models used for specifications are used also for describing the final presentation (i.e., placing the objects onto a timeline).

The main advantage of the operational approach is of being direct. The specification process itself is a clear preview of the final result. This allows the author to have a great control over the single object, and it makes simpler to design and implement authoring systems based on it. This approach has however a major drawback in that the author must have in mind the overall structure of the final presentation—i.e., the exact position of each object for each instant. This is not trivial at all when the amount of involved objects is very large and, as expected in a distributed DL context, they come from many different sources. Moreover, the models used for specification could be very complex, since they must be able to describe also the final presentation, and must be perfectly known by the presentation author.

The constraint-based approach (see, e.g., [4–6]) is based on a clear distinction between presentation specification and its final representation. The structures used by operational systems, although they allow a rich description of both the objects and the events, are in fact not as suitable for the specification and not always author-friendly. The constraint-based approach adopts in general the same strategy for the final representation as the operational one, so that most part of both operational and constraint-based systems adopt the XML-based SMIL language for building the final representation scheme. The main difference is that it introduces a higher-level paradigm for the specification, letting the system translate it into the final form. Such paradigm provides a set of *constraints*, which describe the temporal and spatial relations between objects, independently from the model used to implement the final presentation. A major advantage of the use of constraints is that they are also able to define relations which do not require the knowledge of the exact coordinates of objects, but only of the high-level spatial/temporal relations which must occur among them. For example, a temporal constraint such as $T\_Before(\mathsf{a}, \mathsf{b})$ tells the system to play object a *before* object b, without specifying *how long* before. This improves flexibility and simplifies the task of combining several objects, since only the needed details have to be known in advance. Precise details can also be specified when required, since constraints could be less or more strict.

The main drawbacks of the constraint-based approach are a

THE WORK REPORTED IN THIS PAPER IS PARTIALLY SUPPORTED BY THE EU UNDER THE DELOS NETWORK OF EXCELLENCE ON DIGITAL LIBRARIES (WWW.DELOS.INFO).

0-7803-9332-5/05/$20.00 ©2005 IEEE

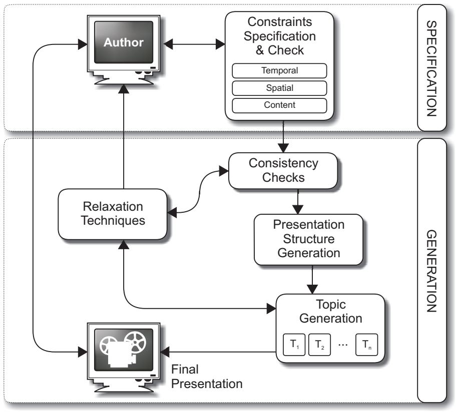

Fig. 1. System architecture

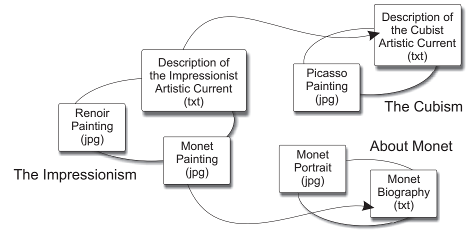

Fig. 2. Interconnected topics

lower degree of control on the final result—since the specification process does not offer a clear projection of the presentation, which is automatically generated by the system in a separate task—and a higher complexity of the final system itself. The system should be able to apply inference rules to determine all the logical implications among the specified constraints, in order to check their consistency and to automatically translate them into a ready-to-play presentation which satisfies all the defined relations.

In this paper, we propose a new authoring approach based on an innovative presentation structure and on a new class of content-based constraints, able to overcome some of the limitations of existing models and systems. Our proposal makes use of a flexible heuristic process, which automatically generates the final presentation. Such innovative features allow the author to easily combine several multimedia objects into a multi-topic presentation, whose different contents can be freely chosen by end users according to their preferences or skill levels.

2. OUR PROPOSAL

The authoring model we propose adopts a constraint-based approach. As previously mentioned, such an approach gives enhanced flexibility to the authoring process, but it requires to perform several tasks in order to check, solve, and translate the specified constraints into a multimedia presentation which satisfies them. The system architecture is depicted in Figure 1.

The Presentation Model Typically, multimedia presentations consist of several media objects, played according to a given temporal sequence which covers one or more topics. Such sequence is usually not modifiable by end users, and it is not possible to freely choose the topics to be shown. Even though a model may support user interactions with the presentation, the easiest way to make it possible for users to choose different contents is to develop specialized presentations for each available topic. Such 'basic' solution has many limitations, since it is difficult to apply whenever there is a large number of topics or, even more, whenever a strong interconnection among them is required.

To overcome such drawbacks, we introduce a model according to which a single presentation consists of several sub-presentations, each representing a *topic* composed of semantically related multimedia objects. All the relations concerning time, layout, and topics can be specified in a single step, so that the presentation author has only to define a set of high-level constraints, used by the system to automatically group objects into topics, instead of developing each sub-presentation individually. Moreover, the relations regarding the topics allow not only to group objects together, but also to define the *connections* among them. Connections are very important since they allow end users to jump from one topic to another one just choosing the connection to follow, according to the topic it leads to.

Example 1 *Let us assume that a simple multimedia database contains three JPEG images, representing paintings by Monet, Renoir, and Picasso, respectively, a JPEG image with a portrait of Monet, and three plain texts containing, respectively, a description of Impressionism and Cubism, and a biography of Monet. The author can group the paintings by Monet and Renoir together with the "Impressionism" text, in a common topic (say, "The Impressionism"), as well as the painting by Picasso and the "Cubism" text (to build "The Cubism" topic). Specific information about Monet can also be grouped into a separate topic ("About Monet"). The author can also define a connection from the painting by Monet to Monet biography, as well as from the "Impressionism" to the "Cubism" text; the defined connections then establish "The Cubism" and "About Monet" topics as logical successors of "The Impressionism" topic (see Figure 2). At execution time, the end user is able to choose the topics to be shown according to his/her interests, as well as the in-depth level (i.e., he/she can browse the detailed information about Monet, or simply skip it).*

Presentation Specification The main issue concerning the development of a multi-topic presentation is that it could require the author to have in mind its overall structure—i.e., the contents of each topic, and the interconnection grid among them. In order to improve the efficiency of the specification process, our approach does not require to define either the presentation structure or the exact spatio-temporal position of the objects composing it. Rather, the specification simply consists of the definitions of high-level relations among objects. Starting from such high-level specifications the system builds both the structure and the interconnection grids, and generates the temporal sequence and the spatial layout of the objects inside each single topic. Such relations are expressed through *constraints* which define the way objects have to be placed in space and time, as well as the way they must be grouped into topics and how they must be connected. Besides temporal constraints, which are based on the qualitative relations defined in [7], and spatial ones, based on those defined in [8], the system supports a new class of constraints called *content constraints*. Content constraints express the semantic relations which make two objects belonging to the same topic (C_Same), to different ones ($C\_Different$), or to two directly connected topics (C_Link).

Presentation Generation The generation process is in charge of three main tasks: *consistency check*, *presentation structure generation*, and *topic generation* (see Figure 1).

The consistency of a presentation is checked by applying compatibility rules to each pair of constraints (also of different kinds) in order to detect inconsistencies. Before performing the check, several inference rules are applied to the initial specification in order to gather the constraints which, even if not explicitly specified, are a logical consequence of the defined ones. If an inconsistency arises, the system applies *relaxation techniques* which reduce the set of constraints until the presentation becomes consistent or, if it is not possible, the author is asked to revise the specification. Other checks are then performed during the subsequent phases to ensure the global consistency of the specification.

The presentation structure generation process builds a consistent presentation structure which reflects the given specification. The underlying structure (which is transparent to the presentation author) is represented by the *presentation graph*, which is a direct graph $G = \langle V, E \rangle$, where each node represents a *presentation topic*, whereas each edge denotes a *connection* between two topics. A presentation graph is consistent if and only if each node is connected to at least another one, because topics represented by nodes with neither incoming nor outgoing edges are impossible to reach during the execution of the final presentation. The generation process always returns a consistent graph (and it is also able to automatically correct some inconsistencies), or the author is asked to revise the specification when it is impossible to generate a consistent one.

The generation procedure consists of four main steps. In the first step, each media object is assigned to a topic according to the given content constraints. The second step builds the connections among topics according to the defined C_Link constraints. In the third step, the consistency of the graph is checked and automatically corrected, whenever possible. The final step performs further checks and returns the final graph $G = \langle V, E \rangle$ or notifies the inconsistency to the author, if the generation process fails. Such procedure tries to make consistent a graph with 'isolated' topics by automatically generating connections which can be inferred from the specification. In particular, when two objects are involved in a T_Before constraint and belong to different topics, the second object must belong to a topic which follows (directly, or passing through n other topics) the one containing the first object, in order to satisfy such constraint. As a consequence, if an isolated topic contains an object b which is involved in a $T\_Before(\mathsf{a}, \mathsf{b})$ relation, it is possible to create a connection from the topic containing a to the isolated one, or even to merge both topics into a single one, if no constraints preventing this restructuring operation have been specified (for example, a $C\_Different$ constraint).

Example 2 *Let us consider the specification in Figure 3. The resulting graph, depicted in Figure 4, is inconsistent, since there is a topic (namely, "The Cubism") with neither incoming nor outcoming edges. The specified T_Before constraint implies that the "Cubism" topic should belong to the same path as the "Impressionism" topic. Since a $C\_Different$ constraint has also been specified, the two topics cannot be merged together: then, an edge connecting the nodes representing the "Impressionism" and the "Cubism" topics is automatically created, making the graph consistent (the resulting graph is the same depicted in Figure 2).*

Finally, topic generation is in charge of building, *for each single topic*, the exact spatial layout and the temporal sequence of the objects belonging to it. The way objects can be placed in space and time depends on the class of constraints defined for them. In particular, two classes of constraints can be identified: *hard constraints*, which impose an exact time reference between two objects (e.g., the $T\_Finishes$ constraint imposes two objects to finish their playout at the same instant), and *soft constraints*, which do not impose an exact time reference (e.g., the $T\_Overlaps$ constraint is satisfied when the playout of the involved objects overlaps, however long the overlapping interval could be).

Temporal placement is carried out by grouping objects into blocks whose members are mutually related through one (or more) hard constraints. Due to the characteristics of such constraints, the temporal location of objects is *fixed* inside each set. The whole blocks can instead be placed into the presentation in a flexible way, which depends on the soft constraints that have been defined upon objects belonging to different blocks. Such flexibility allows authors to apply different criteria when placing blocks in time, so it is possible to generate several different presentations which satisfy the initial specification, according to author- or user-defined parameters, such as the minimal duration.

Spatial placement is carried out by analyzing the space available on the screen at each instant, and partitioning it into *available rectangles* in which the objects can be placed. Since objects can be placed in different available rectangles, each of them satisfying the specified spatial constraints, also the spatial layout can be generated in a flexible way according to different generation parameters. If it is not possible to place a block on the screen without violating any constraint, the system applies the relaxation techniques used during the initial check, in order to make such placement possible. If the process fails, the author is asked to revise the specification; otherwise, the exact spatial and temporal positions of objects inside each single topic are returned, and the generation process is complete.

Obj = {Monet-paint.jpg, Renoir-paint.jpg,
Picasso-paint.jpg, Monet-portr.jpg,
Impressionism.txt, Cubism.txt, Monet-bio.txt}

$Constr$ = { C_Same(Monet-paint.jpg, Renoir-paint.jpg),
C_Same(Monet-paint.jpg, Impressionism.txt),
C_Same(Monet-portr.jpg, Monet-bio.txt),
C_Same(Picasso-paint.jpg, Cubism.txt),
$C\_Different$(Monet-paint.jpg, Monet-portr.jpg),
$C\_Different$(Impressionism.txt, Cubism.txt),
C_Link(Monet-paint.jpg, Monet-bio.txt),
T_Before(Impressionism.txt, Cubism.txt)}

Fig. 3. An example of presentation specification

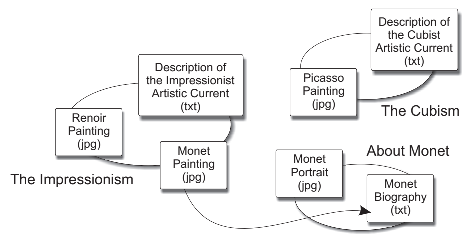

Fig. 4. An example of inconsistent graph

3. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a multimedia authoring approach supporting constraints for building multiple execution flows of the same multimedia presentation, which can then be chosen by end users according to their preferences and skill level. Multimedia objects are described on the basis of their spatial, temporal, and semantic relationships: this information is then used to generate the final presentation through a flexible heuristic process which takes into account the specified constraints. The main difference with respect to similar approaches is that the execution flow is not explicitly defined, but it is built dynamically on the basis of the semantic correlations existing among multimedia objects. Thanks to this feature, we can support multiple execution flows within the same presentation, where each flow corresponds to the different *topics* and/or *sub-topics* into which objects can be grouped.

A prototype implementing our model is now under development. Such prototype is built according to a client-server architecture, and allows authors to store objects and to perform the presentation specification task through a Web interface. All the information concerning constraints, derivation rules, objects, and presentations is stored in a relational database. Finally, a generation module processes such information and builds the different possible instances of the presentation.

We plan to extend our model by supplying personalization features concerning, on one hand, duration requirements and/or preferences, and, on the other hand, the end user's device and network connectivity.

The former feature requires that the duration of execution flows may vary depending on a minimum/maximum time range specified by end users. Thanks to the flexibility of the heuristic generation process, it is possible to freely choose when and where the 'hard' blocks have to be placed inside a topic (when they are not involved in soft constraints with other 'hard' blocks) and/or the temporal and spatial distance between them (within the range which satisfies the qualitative soft contraints in which they are involved). Several strategies can thus be used when placing 'hard' blocks in time and space in order to obtain several 'versions' for each topic, each one satisfying the presentation specification. As the duration of the alternative versions may be different, it is possible, by combining such versions properly, to modify the duration of each execution flow without the need of relaxing the specification or of reducing the quality of objects. This allows the system to perform the required temporal optimization with no impact on the presentation quality. If a more enhanced optimization is needed, it is also possible to enforce additional optimization strategies concerning both the temporal and the semantic characteristics of objects. More precisely, the most relevant portion of each object is specified, corresponding to the 'condensed' view of the object itself. The whole or the condensed view of the object is then displayed depending on the chosen time range. Additionally, each object can be associated with a 'weight', stating its relevance with respect to the corresponding topic(s); objects are then chosen or discarded by evaluating such metadata.

The latter feature aims at supporting multimedia adaption in order to allow end users to display efficiently and effectively multimedia presentations independently from their devices and network connectivity. This requires mainly two classes of adaptation techniques. The former concerns the placement and dimension of multimedia objects, which must be adapted to the display characteristics of the device. The latter concerns the 'quality' of objects, which must be adapted to, e.g., the available network bandwidth, and the color depth and resolution supported by the device. Note that multimedia adaptation cannot be enforced by simply decreasing objects low-level features, since this may often result in making objects unusable. Thus, it is necessary not only to specify scalability information for each object, but also to define rules in order to maintain execution flows consistent even though some objects are discarded because of their scalability limitations.

Finally, a further extension concerns the adoption of the MPEG-7 and MPEG-21 standards for representing the structure and the temporal, spatial, and semantic constraints of objects. MPEG-7 and MPEG-21 are becoming the standard technologies in Audio-Visual DL (AVDL) for encoding metadata. Currently, they are widely used in AVDL systems, and several improvements have been proposed—such as the adoption of semantic extensions to XML and the support for ontologies, which would permit the full integration of AVDL systems in the Semantic Web framework. It is then important to support such standards in order to provide interoperability. Moreover, they supply features which can be directly used in our approach. MPEG-7 can be used to describe the structure of each object, and its temporal, spatial, and semantic characteristics, whereas MPEG-21 can be used to define the presentation as a structured set of multimedia objects with semantic relationships, to enforce personalization with respect to content and duration requirements/preferences, and to adapt object characteristics and placement to end users' access features.

4. REFERENCES

[1] L. Hardman, G. van Rossum, and D. C. A. Bulterman, "Structured multimedia authoring," in *Proc. of the 1st ACM Int. Conf. on Multimedia (MULTIMEDIA93)*, 1993, pp. 283–289.

[2] P. Ackermann, "Direct manipulation of temporal structures in a multimedia application framework," in *Proc. of the 2nd ACM Int. Conf. on Multimedia (MULTIMEDIA94)*, 1994, pp. 15–58.

[3] S. Boll and W. Klas, "ZyX – A semantic model for multimedia documents and presentations," in *Proc. of the 8th IFIP Conf. on Data Semantics (DS-8)*, 1999, pp. 188–209.

[4] K. S. Candan, B. Prabhakaran, and V. S. Subrahmanian, "CHIMP: A framework for supporting distributed multimedia document authoring and presentation," in *Proc. of the 4th ACM Int. Conf. on Multimedia (MULTIMEDIA96)*, 1996, pp. 329–340.

[5] L. Tardif, F. Bes, and C. Roisin, "Constraints for multimedia documents," in *Proc. of the 2nd Int. Conf. and Exhibition on The Practical Applications of Constraint Techniques and Logic Programming (PACLP00)*, 2000.

[6] J. Geurts, J. van Ossenbruggen, and L. Hardman, "Application-specific constraints for multimedia presentation generation," in *Proc. of the Int. Conf. on Multimedia Modeling (MMM01)*, 2001, pp. 247–266.

[7] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[8] J. Z. Li, M. Tamer Özsu, and D. Szafron, "Spatial reasoning rules in multimedia management systems," in *Proc. of the 3rd Int. Conf. on Multimedia Modeling (MMM96)*, 1996, pp. 119–133.