

Representing and Reasoning with Temporal Constraints in Multimedia Presentations

S. Adali
Rensselaer Polytechnic Institute
sibel@cs.rpi.edu

L. Console, M.L. Sapino, M. Schenone
Università di Torino
(lconsole, mlsapino, schenone)@di.unito.it

P. Terenziani
Università del Piemonte Orientale
terenz@di.unito.it

Abstract

With the growing use of computers to create technical, business and personal presentations, there has been a huge increase in the number of multimedia presentation tools out in the market. Thus, over the coming years, there will be a growing need to build the ability to query rapidly databases consisting of interactive multimedia presentation.

The MPA algebra proposed in [1] provides a specification of how primitive algebraic operations on interactive multimedia presentation can be implemented. In this paper we define a temporal extension of MPA dealing with temporal constraints in the representation and in the queries. In particular, we point out the different types of (explicit and implicit) temporal constraints within a multimedia presentation and define a suitable framework in which the different types of constraints can be represented and queries (regarding also the temporal aspects) can be answered efficiently.

1. Introduction

Over the last few years, there has been a large increase in the number of interactive multimedia presentations created by many different organizations. These presentations are created by using multiple different multimedia packages and typically presented using Internet browsers, plugins and special purpose packages. The concept of an *interactive multimedia presentation database (IMP-DBMS)* has only been recently introduced [8]. However, to build a full fledged IMP-DBMS mirroring the breadth and power of the relational databases, many major advances are needed. One of these challenges is that of building a version of the relational algebra to query and manipulate IMP-DBMSs. In [1], some of the authors proposed an algebra called MPA (Mul-

timedia Presentation Algebra) to query multimedia presentation archives.

The algebra proposed in [1] provides a *specification* of how primitive algebraic operators on IMPs can be implemented. In this formalism, IMPs are intuitively represented as trees of nodes as follows. The nodes of such a tree are labeled with non interactive presentations [2, 6, 7, 9, 10, 14], that are sets of multimedia objects, on which both temporal and spatial constraints are defined. Each arc in the tree represents a possible interaction. An interaction causes a node to be transformed into another node. Thus, when a single node is being presented, the user cannot interact with it, but he can interact with a node and select which other node to navigate to. In its simplest form, a node can be a static HTML page, with different interactions corresponding to different hyperlinks. Alternatively, a node may consist of a video clip, with different nodes representing fast forward, rewind, pause operations which transform the presentation. In short, the paths in the tree represent different possible playouts of the presentation. IMPs represent the logical structure of the set of all possible playouts, and may be infinite. Notice however that the underlying physical structure is a graph.

The algebra we presented in [1] consists of different operators to split, modify and rearrange multimedia presentations. Almost all of these operations, however, are based on the possibility of expressing selection conditions that determine in a given presentation which paths, nodes and objects to operate on. Within this context, in [1] temporal constraints were only considered in a very marginal way (see the concluding section) and, in particular, there was no way of operating on objects and nodes on the basis of their temporal properties. Of course, this is a very severe limitation in many applications. In this paper, we extended the approach in [1] to deal with temporal constraints both in the representation (definition) and in the selection (query) of in-

teractive multimedia presentations, discussing the following contributions.

- First, we analyse the role of temporal constraints in multimedia presentations. In particular, we deal with different types of (explicit and implicit) temporal constraints: *intra-node* temporal constraints, that relate different objects within a presentation node, and *inter-nodes* temporal constraints, that express constraints between consecutive nodes. Taking into account space and time complexity issues, we define a suitable temporal framework, and a temporal reasoning algorithm to check consistency. We prove that, although for the sake of computational efficiency such an algorithm operates locally on the objects of each node, it is complete.
- Focusing on selection conditions, we extend the (atemporal) query language defined in [1], to a temporal dimension: we allow temporal queries with both MUST and MAY modality, and introduce hypothetical temporal queries.
- Finally, we discuss some complexity results, and show that the evaluation of temporal conditions is independent of the size of the whole presentation, but only depends on (i) the number of objects explicitly mentioned in the query, and (ii) the number of nodes in the path that the query has to be evaluated against.

On the other hand, in this paper we do not take into account the temporal extension of the operators of the algebra in [1], which can be defined on the basis of the temporal framework discussed in this paper, and is the goal of our future work.

2. Temporal Interactive Multimedia Presentations

Intuitively, a temporal interactive multimedia presentation (TIMP) is a collection of presentation nodes where each node is a single non-interactive multimedia presentation involving multiple objects, visible and/or audible. An interaction reflects a possible transition from one node to another using any type of input, such as the push of a button, a voice command, hitting a key, or entering text in a search field. To define a TIMP, we need some intermediate definitions.

We will assume the existence of some set of *objects*, each with an associated *type* (e.g. gif, tiff, mpg, etc.). Each object may have zero or more attributes — we will write $o.A$ to refer to the value of attribute A of object o . In addition, we assume the existence of *temporal variables* of the form $\text{Start}(X)$ and $\text{End}(X)$ where X is a symbol denoting an object or a presentation node. These variables

denote the start and end times of the presentation of objects and nodes, and are used in temporal constraints as described below.

Definition 2.1 (Presentation Node) A presentation node N is a pair $\langle \text{CONT}_N, \text{INT}_N \rangle$ where CONT_N is the content of node N and INT_N is the set of all possible interactions on the content of node N . The content component CONT_N is a pair $\langle O_N, \text{TC}_N \rangle$, where O_N is the set of objects belonging to the node (to be presented in a non-interactive way), and TC_N are the *intra-node* temporal constraints on the temporal variables denoting start and end times of objects in O_N .

The set of *intra-node* temporal constraints indicate the desired temporal properties of the objects located at that node, in general, relative to the starting time of the presentation node. The constraints can state the temporal duration of an object, or specify the relative temporal positions of two objects, or the delays between the start and end points of their presentation. In general, these constraints do not contain absolute dates since multimedia presentations are intended to be presented several times, and not once and forever at a specific time point. Typically, a multimedia application displaying a presentation node will find a solution to the *intra-node* constraints and show/play the objects of this node according to this solution. We discuss the syntax of these constraints in detail in Section 2.1.

Interactions in INT_N are functions from nodes to nodes; when an interaction $i \in \text{INT}_N$ is applied to the node N , another node N' is returned. N' 's content is obtained from N 's content, according to the transformation process that is encoded in the function i . We use the notation $N \rightarrow^i N'$ to denote this transformation. INT_N contains as many interactions as the possible playouts starting from the node N . (see [1] for more details on this aspect).

From the temporal point of view, an interaction between a parent node $N1$ and a child node $N2$ imposes an *inter-node temporal constraint* between $N1$ and $N2$. These constraints represent the delay between the ending objects of a node and the starting objects of the following node, due to an interaction with the user. In many cases, one could assume that this delay is null. On the other hand, in the general case, such a delay depends on many different factors, including the interaction time with users, the time needed in order to load the objects of the following node (which, in turn, may depend on the work-load of the system) and so on. Making hypotheses on the amount of such delays is a very important operation at query time, when, presumably, the user has specific knowledge on the specific environment in which the multimedia presentation is to be used. The concepts introduced above are formally represented in the following notion of **TIMP-Space**, Temporal Interactive Multimedia Presentation Space.

Definition 2.2 Temporal Interactive Multimedia Presentation (TIMP) Space A temporal interactive multimedia presentation space (or TIMP_Space for short) is a *labelled tree* $Tree$ such that each node N in $Tree$ where $N = \langle CONT_N, INT_N \rangle$ is a presentation node and for all $i \in INT_N$, N has a single child node N' in $Tree$ such that $N \rightarrow^i N'$. Each edge in the tree is labelled with a constraint involving the temporal variables denoting the start time of the child presentation node relative to the end time of parent presentation node. These labels constraint the minimum and maximum duration of an interaction¹. The root contains a special object *start* in its content, and all the leaf nodes contain only special *end* objects and no interactions.

Example 2.1 In this example, we consider a multimedia application that takes viewers through a “Guided Tour of National Gallery”. The guided tour includes pictures of paintings and sculptures, various audio and video clips describing these works of art. Each room in the gallery represents a different node in a TIMP. The objects in a node are the files for the pictures and clips. The temporal constraints in a node represent a script, i.e. a sequence in which to show the pictures of art work, synchronization of audio clips with the relevant picture and delays between one picture to the next. In addition, a small video is shown at the entrance to each room to give the historical context of the works within that room. Interactions advance the user to the next room. The user can choose the quickened version of the presentation for a room (“QUICK TOUR”) or the standard version (“NEXT ROOM” in Figure 1). Figure 1 shows a part of a multimedia presentation. For the sake of clarity, we did not explicitly represent the temporal constraints in the nodes and on the arcs of the tree.

2.1 Representing temporal constraints in a multimedia presentation: a tree of STP frameworks

Let us now consider how to represent *intra-node* and *inter-node* temporal constraints in a multimedia presentation. As discussed above, we use the same type of constraints for both cases. However, intra-node constraints involve temporal object variables and inter-node constraints involve temporal node variables. For the sake of simplicity, and in order to have computational efficiency, we focused our attention to those constraints which can be mapped onto sets (conjunctions) of bounds on difference constraints [12, 13]. Bounds on differences are linear disequalities of the form $c \leq X - Y \leq d$. As discussed, e.g., in [13, 11], these constraints have a temporal interpretation in which the variables X and Y are two time points

¹The pair $\langle 0, \infty \rangle$ can be used in case such a duration is completely unknown.

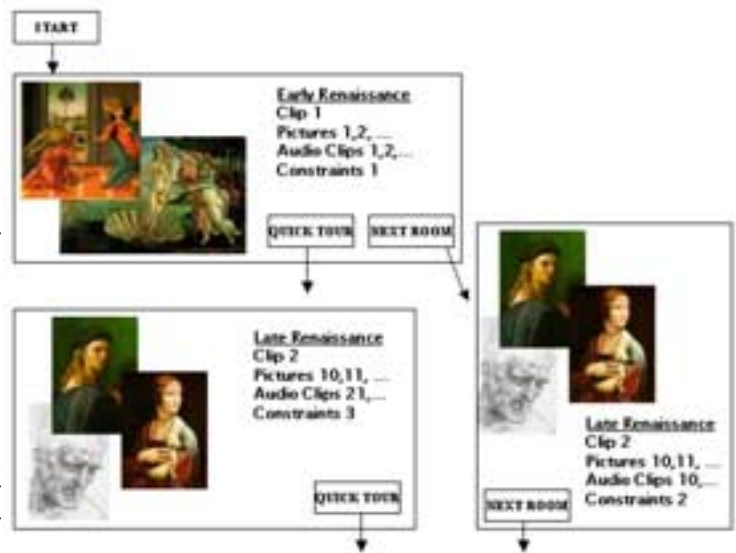


Figure 1. The presentation for the guided tour in Example 2.1.

(representing, e.g., the starting and/or ending points of two events) and difference is interpreted as temporal distance. Thus, conjunctions of bounds on differences also called STP frameworks [13] can easily represent:

- exact or approximate durations, by stating the distance between the endpoints of an event;
- exact or approximate delays between two time points;
- qualitative relations between time points, excluding disequality (thus, only continuous pointizable qualitative temporal constraints can be dealt with [18]).

Notice that some disjunctive qualitative constraints between time intervals (e.g., I_1 before or after I_2) cannot be mapped onto bounds of differences, but this is a reasonable assumption in many applications, as discussed in [17]. The advantage of using the STP framework is that its consistency can be checked in a time cubic in the number of variables (time points) by using an all-shortest-path algorithm. Moreover, the all-shortest-path algorithm can also produce a compact representation of all the solutions (assignments to the variables which satisfy all the constraints) called *minimal network*, which contains, for each pair $\langle X, Y \rangle$ of variables, the maximal admissibility range $[c,d]$ that includes all and only the values for $X-Y$ which satisfy all the constraints in the knowledge base.

Thus we chose to use an STP framework to represent **intra-node temporal constraints**. In particular, we adopted a separate STP for each node, in order to keep the

constraints between objects of different nodes separate (in addition to avoiding the treatment of meaningless temporal constraints, this approach has dramatic advantages in terms of space complexity). In particular, we associate with each node N , the minimal network of the temporal constraints between the objects in N , and the start and end times of the presentation of node N itself. Maintaining the minimal network provides crucial advantages at selection time, as described in Section 4

In order to manage the *inter-node temporal constraints*, we explicitly maintain the tree of the nodes in the presentation, considering that, if a node $N2$ is the son of a node $N1$ in the tree, we have the implicit temporal constraint $0 < \text{Start}(N2) - \text{End}(N1)$. Thus, the overall representation of the temporal constraints is a tree of minimal networks $\{Tree, \{MN1, \dots, MNk\}\}$, where the arcs in tree are labeled with the minimal and maximal distance between the end of the parent node and the start of the son node (in general, since such delays depend on context and run time, only the precedence constraint is maintained).

In our multimedia application, we do not have the minimal network of the whole set of constraints. However, the consistency of the whole set of constraints regarding nodes and objects of a multimedia presentation can be checked without a global propagation involving all the time points and constraints in the multimedia.

Example 2.2 Recall the guided tour example given in Example 2.1. Suppose a particular room (e.g., room $r13$) in this application contains three paintings with objects o_1, o_2, o_3 denoting the pictures of such paintings, a_1, a_2, a_3 denoting the audio clips about these paintings, and v is a video clip about the room in general. The following is a partial set of constraints for this room (time unit is seconds):

1	\leq	$\text{start}(o1) - \text{end}(v)$	\leq	2
10	\leq	$\text{start}(o2) - \text{end}(o1)$	\leq	20
2	\leq	$\text{start}(o3) - \text{end}(o2)$	\leq	4
1	\leq	$\text{start}(a1) - \text{start}(o1)$	\leq	2
4	\leq	$\text{start}(a2) - \text{start}(o2)$	\leq	6
20	\leq	$\text{end}(o3) - \text{start}(o3)$	\leq	20
0	\leq	$\text{start}(r13) - \text{start}(v)$	\leq	0
0	\leq	$\text{start}(a3) - \text{start}(o3)$	\leq	0
0	\leq	$\text{end}(a3) - \text{end}(v)$	\leq	0
10	\leq	$\text{end}(o1) - \text{start}(o1)$	\leq	10
15	\leq	$\text{end}(o2) - \text{start}(o2)$	\leq	15

Property 2.1 *The temporal constraints $\{Tree, \{MN1, \dots, MNk\}\}$ of a multimedia presentation are consistent if and only if $MN1, \dots, MNk$ are consistent independently of each other.*

The problem of selecting and projecting objects nodes and paths on the basis of their atemporal properties (e.g., their types, their components, the length of a path between

two nodes), has already been investigated in [1]. In this paper, we concentrate on the issues concerning temporal properties of the object, nodes and paths, and show how some temporal operators and temporal queries can be easily and effectively integrated in the already existing atemporal framework².

3. Selection Conditions over TIMPS

In this section, we describe different classes of selection conditions over TIMPs. The main aim of the selection conditions is to locate items of interest in a TIMP database. These conditions will be used in future work to define algebraic operators that find and alter parts of existing temporal interactive multimedia presentations based on user queries.

Object selection conditions. These are defined over objects appearing in presentations, on the basis of their static attributes, and temporal attributes. “Select all the videos, in the presentation *Guided tour of National Gallery*, whose topic is Italian Artists of the XIX Century, and last more than 2 minutes” is an example of an object selection condition.

Node selection conditions. These are defined over nodes (frames) appearing in the presentation, on the basis of the objects they contain, and of the relationships among them. “Select all the frames, in the presentation *Guided tour of National Gallery*, that contain at least a picture-object, and an audio object, that must be presented simultaneously” is an example of a node selection condition.

Path selection condition These are defined over paths in the presentation tree, representing possible payouts. “Select all the payouts, in the presentation *Guided tour of National Gallery*, that are sequences of at least 5 nodes such that (i) each node contains at least a picture-object, and an audio object, to be presented simultaneously, and (ii) the payout time, i.e., the time between the start instant of the first node, and the end instant of the last node, is less than 20 minutes.” is an example of a path selection condition.

As shown by the examples above, in the approach described in this paper selection conditions may consist of both a temporal and an atemporal part. In the following, we discuss such two parts separately, starting from a brief recall of atemporal conditions, as given in [1].

² Actually, the temporal and spatial constraints associated to the objects in the multimedia presentation were already present in [1]. But in that paper, no temporal attribute could be queried, and no selection condition based on temporal aspects could be defined.

3.1 Atemporal object selection conditions

The first type of expressions we describe are the *atemporal object selection conditions*. These conditions contain variables (denoted O, O_1, O_2, \dots) that range over multimedia objects stored at different nodes. In addition, we assume the existence of a set of variables ranging over base types, and over temporal attributes of objects.

An *atomic atemporal selection condition* on O is defined as follows: (i) If t_1, t_2 are terms (variables or objects) of the same type, then $t_1 = t_2$ and $t_1 \neq t_2$ are selection conditions. (ii) If t_1, t_2 are terms (variables or objects) of a type that has an associated partial ordering \leq on its objects (e.g. *real, integer, string*), then $t_1 \Re t_2$ is a selection condition, where \Re is one of $\{<, \leq, >, \geq\}$.

(Atemporal) Weak Selection conditions may now be inductively defined as follows: (i) every atomic atemporal selection condition is a weak selection condition, (ii) if woc_1, woc_2 are weak object selection conditions, then so are $(woc_1 \wedge woc_2)$ and $(woc_1 \vee woc_2)$. An *object selection condition* is a weak object selection condition containing exactly one object variable symbol in it.

Example 3.1 To find all audio objects that are on the subject “Italian Artists of the XIX Century”, we can write the following object selection condition: $O.type = \text{“audio”} \wedge O.subject = \text{“Italian Artists XIX Century”}$.

3.2 Atemporal node selection condition

An object selection condition describes a set of objects regardless of which node they may appear in. Node selection conditions, on the other hand, describe nodes based on their contents. To define atemporal node selection conditions, we assume the existence of a set of variables called node variables (denoted N, N_1, N_2, \dots). A *node term* is either a node, or a node variable.

If N_1, N_2 are nodes in a path of a TIMP, then we use the notation $N_1 \ominus N_2$ to denote $level(N_1) - level(N_2)$ where $level(N_i)$ is the level of node N_i in the TIMP (assuming the root to be at level 0). If two nodes do not lie on a common path, then \ominus is undefined for these nodes. Weak (atemporal) node selection condition is defined as follows:

- If O is an object variable, N is a node variable, and o is an object identifier, then $O \in N$ and $o \in N$ are weak node selection conditions.
- If N_1 and N_2 are node terms, then $N_1 \ominus N_2 \langle op \rangle c$ is a weak node selection condition, where $\langle op \rangle \in \{<, \leq, >, \geq\}$ and c is an integer.
- If wnc is a weak node selection condition and woc is a weak object selection condition then $wnc \wedge woc$ is a weak node selection condition.

- If wnc_1 and wnc_2 are weak node selection conditions then so are $wnc_1 \vee wnc_2$, and $wnc_1 \wedge wnc_2$.

A *node selection condition* is a weak node selection condition that contains exactly one node variable.

In particular, a weak node selection condition of the form $N_1 \ominus N_2 \geq k$ for some positive number k can be read as “node N_1 is a successor of node N_2 , and there exist at least $k - 1$ nodes between the two nodes.”

Example 3.2 To find all rooms containing both XIX century and XX century paintings by Italian artists, we can write the following node selection condition: $O1 \in N \wedge O2 \in N \wedge O1.subject = \text{“Italian Artists XIX Century”} \wedge O2.subject = \text{“Italian Artists XX Century”}$.

3.3 Atemporal path selection conditions

Finally, we assume the existence of a set of variables called path variables (denoted \wp_1, \dots). A path term is either a path variable or a path. If N is a node term and \wp is a path term then $N \in \wp$ is a weak path condition. If wpc is a weak path condition and wnc is a weak node condition, then $wpc \wedge wnc$ is a weak path condition. The conjunction and disjunction of weak path conditions is also called a weak path condition. A *path selection condition* is a weak path selection condition containing exactly one path variable. The \ominus operation, conjunction and disjunction between node conditions makes it possible to express all regular expressions between node variables for describing paths. Since node/path selection conditions may contain weak object selection conditions, it is possible to talk about the relationship of two objects in a single node as well.

3.4 Temporal Conditions

Temporal conditions on a multimedia presentation can be distinguished according to different parameters, including: (i) type of the temporal condition (ii) modality of the temporal condition selection.

- (i) As regards to the type of the temporal condition, it may concern the (exact or approximate) dates when the entities occur, their duration, the delays between their end points, and qualitative relations (e.g., precedence) between them. As discussed, e.g., in [4, 11], all these conditions can be expressed as conjunctions of bounds on differences constraints involving temporal constraints. On the other hand, to deal with time in multimedia selection conditions, it is useful to distinguish between *unary* and *binary* temporal constraints. Unary temporal constraints regard one entity (e.g., one object) at a time. In particular, they may concern

the *dates* of the starting and ending points of the entity, or its *duration*. Binary temporal constraints relate two different entities (nodes or objects). These constraints concern *qualitative relations* (e.g., precedence) between starting and ending points of entities or (exact or approximate) *delays* between them.

$\langle TC \rangle ::= \langle \text{Unary-TC} \rangle \mid \langle \text{Binary-TC} \rangle$

$\langle \text{Unary-TC} \rangle ::= \text{Date} \mid \text{Duration}$

$\langle \text{Binary-TC} \rangle ::= \text{Qual-Rel} \mid \text{Delay}$

In particular, for the moment we only consider conjunctive temporal conditions, which might be expressed as sets of atomic conditions.

- (ii) Since in almost all practical cases the temporal constraints between entities in the model are such that the exact temporal locations of entities is not known, it is important to distinguish on whether the temporal conditions must be intended as necessary conditions (i.e., conditions that necessarily hold, given the temporal constraints in the model; called **MUST** queries in the following) or possibility conditions (i.e., conditions which may hold, given the temporal constraints in the model, or, in other words, which are consistent with the constraints in the model; called **MAY** queries in the following) (see [17, 5]).

In particular, **MAY** queries are important in *path selection conditions*, where the possibly unpredictable time of user interaction has to be taken into account.

$\langle \text{ModalOP} \rangle ::= \text{MUST} \mid \text{MAY}$

A further type of temporal condition is very important in many practical applications: hypothetical condition. In fact, as discussed in section 2, the delays due to the interactions cannot be fixed a-priori (when building the model). On the other hand, the user might want to select paths on the basis of certain hypotheses on the duration of such delays. E.g., “Select all the playouts, in the presentation *Guided tour of National Gallery*, that are sequences of at least 5 nodes such that and the overall playout time is less than 20 minutes, *assuming that each interaction requires at most 30 seconds.*”

We can now define the temporal conditions used in object, node and path selection conditions.

As discussed above, (atemporal) selection conditions on objects select objects independently of each other. Hence, we define object temporal selection conditions as a **MUST** or a **MAY** query on a set (conjunction) of unary temporal conditions. On the other hand, node and path selection conditions relate objects and nodes in **TIMP**'s paths. Thus, in temporal node/path selection conditions, we allow also binary temporal constraints relating different entities (objects and/or nodes), and hypothetical temporal queries.

$\langle \text{SET-Unary-TC} \rangle ::= \langle \text{Unary-TC} \rangle \mid \langle \text{Unary-TC} \rangle,$

$\langle \text{SET-Unary-TC} \rangle$

$\langle \text{SET-TC} \rangle ::= \langle \text{TC} \rangle \mid \langle \text{TC} \rangle, \langle \text{SET-TC} \rangle$

$\langle \text{Object-Selection} \rangle ::= \langle \text{Modal-op} \rangle \langle \text{SET-Unary-TC} \rangle$

$\langle \text{Node-Path-Selection} \rangle ::= \langle \text{Modal-op} \rangle \langle \text{SET-TC} \rangle \mid$

$\langle \text{Modal-op} \rangle \langle \text{SET-TC} \rangle \text{ IF } \langle \text{SET-TC} \rangle$

Example 3.3 To find all audio objects on the Italian Artists whose duration is at least 2 minutes, we can use the following temporal selection condition: $O.type = \text{“audio”} \wedge O.subject = \text{“Italian Artists”} \wedge \text{MUST}(120 \leq end(O) - start(O))$.

To find all rooms containing a presentation of pictures by the same artist such that the total presentation can exceed 5 minutes, we can use the following condition: $O1 \in N \wedge O2 \in N \wedge O1.artist = O2.artist \wedge \text{MAY}(300 < end(O2) - start(O1))$.

4. An efficient management of temporal selection conditions

In this section, we sketch the algorithms we defined in order to deal with selection conditions, with specific attention on the temporal issues. In particular, we show how, taking advantage of locality properties on bounds on differences [3], we obtained an efficient treatment of temporal conditions.

A first important issue concerns the order in which the temporal and atemporal parts of the conditions have to be dealt with. In principle, since temporal and atemporal parts of the conditions are orthogonal (i.e., they concern completely independent attributes of the objects/nodes), one could choose to start from temporal selection. However, using first the atemporal conditions to select the entities (objects/nodes) and then applying the temporal conditions to the selected entities is a better strategy, which allows temporal reasoning to be performed on specific instances of entities (and not on variables), as in usual AI approaches to temporal reasoning (see, e.g., [18]).

4.1 Selection on objects

The temporal condition in object selection only regards objects in isolation and is used as a second filter on the objects which satisfy the atemporal condition. Thus, in the algorithm, we check temporal conditions independently of each other.

Object_Selection

For each node N in the presentation do

1. Select all objects in the node which satisfy the atemporal object selection condition. Let O be the set of the selected objects;

2. For each object $o \in O$ check the temporal condition on o . Return the set O' of objects which satisfy this check.

As regards the application of temporal constraints to the objects in O , the algorithm applies the constraints on the basis of the description of the objects.

Example 4.1 Given Example 2.2, let us consider the following query:

“Select all paintings by Leonardo whose presentation lasts more than 16 minutes”, which can be expressed as follows:

$X.type = picture \wedge X.author = Leonardo \wedge MUST(16 \leq end(X) - start(X))$.

Limiting the attention to room 13 (see figure 1), and to the constraints in example 2.2, the atemporal condition selects two objects o_2 and o_3 ; however only o_3 satisfies the temporal constraints, and is thus returned as the result of the query.

Maintaining the minimal network of the temporal constraints within each node is advantageous as far as the temporal check at step 2 above is concerned. In fact, the check of the selection temporal condition on a given object o can be simply performed as a retrieval of the maximal admissibility range concerning the endpoints of o from the minimal network, and a comparison of such a range with the conditions in the query. Thus, step 2 above can be performed in linear time on the objects in O .

4.2 Selection on nodes and paths

In the case of nodes and paths, temporal conditions are more complex, since they may concern also binary temporal constraints (and hypothesis on the temporal extent of objects, nodes and interactions). This means that, in such a case, temporal conditions cannot be checked on objects and nodes independently of each other, as in Object_Selection above.

Node-Path_Selection

For each path P in the presentation do

1. Select all objects and nodes which satisfy the atemporal selection condition. Let ON be the set of the selected objects and nodes
2. Apply the temporal condition on the objects and nodes in ON , and check whether it is satisfied. Return all nodes and objects in ON that satisfy the temporal condition.

Notice that, in the case of temporal conditions on nodes and paths, the application of the temporal constraints to the objects in ON is not a trivial process. In fact, it involves

imposing binary temporal constraints on correlated pairs of objects/nodes only. Maintaining the minimal network of the temporal constraints within each node is advantageous also as far as the temporal check at step 2 above is concerned. In particular, Property 4.1 holds.

Property 4.1 *Given the temporal constraints $\{Tree, \{MN1, \dots, MNk\}\}$ of a multimedia presentation, a temporal condition of the forms discussed in section 3 can be checked, for each path selected at step 1, in a time at most cubic in the number of the objects and nodes selected by the atemporal part of the query, plus the number of nodes between selected nodes/objects in that path (plus 1, due to the reference time point for the whole set of temporal constraints), with no loss of information (i.e., giving the same result that would be obtained by performing temporal reasoning on all the objects and nodes in the presentation).*

Property 4.1 is very important, since it allows one to localize temporal reasoning on the objects and paths explicitly mentioned in the query, independently of the dimension of the whole multimedia presentation.

The proof of property 4.1 is based on the theoretical results in [3], and extends it to consider partitioned minimal networks.

5. Conclusions

In this paper, we proposed an approach to deal with temporal constraints both in the representation and in the selection of multimedia presentations. In particular, as regards the representation of temporal constraints, we

- pointed out the different types of (explicit and implicit) temporal constraints within a multimedia presentation;
- defined a suitable temporal framework in which the different types of constraints can be represented. In defining such a framework, we took into account space and time complexity issues (see below);
- defined a temporal reasoning algorithm to check consistency. We proved that, although for the sake of computational efficiency such an algorithm operates locally on the objects of each node, it is complete.

As regards temporal (and atemporal) selection of multimedia presentations (query time), we

- defined a suitable query language to express temporal conditions on objects and nodes in the presentations. In particular, we distinguished between **MUST** and **MAY** selections, and took into account also hypothetical queries;

- defined efficient procedures to deal with temporal and atemporal selection. In particular, in our approach, temporal reasoning to check temporal conditions is performed in a time which is independent of the dimension of the whole presentation (being dependent only on the number of objects explicitly mentioned in the query, and of the nodes in the corresponding path), with no loss of information.

In this paper, we extended the approach dealing with multimedia presentation in [1] to consider temporal constraints both in the representation and in the selection conditions. In fact, in [1] only *intra-node* temporal constraints could be represented, and temporal reasoning issues were almost neglected, since temporal conditions are not defined, so that only atemporal conditions could be used in the selection of objects, nodes and paths.

Lee et. al. [15] introduce a language to query and construct non-interactive multimedia presentations. In this paper, temporal properties of objects are defined using next/until constructs that describe sequential and periodic ordering of objects in time. However, queries involving duration and different modalities are not present in this data model and the query language.

The efficient treatment of temporal conditions discussed in Property 4.1 is based and on the locality properties in query answering on bounds on differences proved by some of the authors in [3]. However, [3] proved the possibility of localizing query answering to the entities explicitly mentioned in the query, given the overall minimal network for the whole set of temporal constraints (bounds on differences). In this paper, we extended such a result to the case of a labelled tree of minimal networks. This has the further advantage that we can keep temporal constraints local to each node, thus achieving in a natural way a useful partitioning of temporal constraints (with the obvious advantages in space complexity).

References

- [1] S. Adali, M.L. Sapino, and V.S. Subrahmanian. "An Algebra for Creating and Querying Multimedia Presentations" ACM Multimedia Systems Journal - Special Issue on Multimedia Presentations, to appear.
- [2] E. Bertino, E. Ferrari. (1998) *Conceptual Temporal Models for Multimedia Data*, accepted for publication in IEEE Trans. on Knowledge and Data Engineering (to appear).
- [3] V. Brusoni, L. Console, and P. Terenziani. On the computational complexity of querying bounds on differences constraints. *Artificial Intelligence*, 74(2):367–379, 1995.
- [4] V. Brusoni, L. Console, B. Pernici, and P. Terenziani. Later: an efficient, general purpose manager of temporal information. *IEEE Expert*, 12(4):56–64, 1997.
- [5] V. Brusoni, L. Console, P. Terenziani, and B. Pernici. Qualitative and quantitative temporal constraints and relational databases: Theory, architecture, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 11(6): 948–968, 1999.
- [6] M.C. Buchanan and P.T. Zellweger (1993) *Automatic Temporal Layout Mechanisms* ACM Multimedia 93, pp. 341-350, 1993.
- [7] M.C. Buchanan and P.T. Zellweger (1993) *Automatically Generating Consistent Schedules for Multimedia Documents*, ACM/Springer-Verlag Journal of Multimedia Systems, vol. 1, no. 2, 1993.
- [8] K.S.Candan, E. Lemar, and V.S. Subrahmanian. (1997) *Management and Rendering of Multimedia Views*, Proc. 1998 Intl. Workshop on Multimedia Information Systems, Sep. 1998. Springer Verlag Lecture Notes in Computer Science.
- [9] K.S. Candan, B. Prabhakaran and V.S. Subrahmanian. (1996) *CHIMP: A Framework for Supporting Multimedia Document Authoring and Presentation*, Proc. 1996 ACM Multimedia 1996 Conference, Boston, MA, Nov. 1996.
- [10] K.S. Candan, B. Prabhakaran, and V.S. Subrahmanian (1996) *Retrieval Schedules Based on Resource Availability and Flexible Presentation Specifications*, Accepted for publication in ACM-Springer Multimedia Systems Journal.
- [11] L. Console and P. Terenziani. Efficient processing of queries and assertions about qualitative and quantitative temporal constraints. *Computational Intelligence*, 15(4):442–465, 1999.
- [12] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32:281–331, 1987.
- [13] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.
- [14] M.Y. Kim and J. Song (1995) *Multimedia Documents with Elastic Time*, ACM Multimedia Conference '95, 1995.
- [15] T. Lee, L. Sheng, T. Bozkaya, N. Balkir, M. Özsoyoglu, G. Özsoyoglu (1999) *Querying Multimedia Presentations Based on Content*. Transactions on Knowledge and Data Engineering, 11(3) pp. 361-385

- [16] V.S.Subrahmanian. (1998) "Principles of Multimedia Database Systems", Morgan Kaufmann
- [17] P. VanBeek. Temporal query processing with indefinite information. *Artificial Intelligence in Medicine*, 3:325–339, 1991.
- [18] L. Vila. A survey of temporal reasoning in artificial intelligence. *AI Comm.*, 7(1):4–128, 1994.