A Multimedia Presentation Algebra

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

M.L. Sapino
Università di Torino
mlsapino@di.unito.it

V.S. Subrahmanian
University of Maryland
vs@cs.umd.edu

Abstract

Over the last few years, there has been a tremendous increase in the number of interactive multimedia presentations prepared by different individuals and organizations. In this paper, we present an algebra for querying multimedia presentation databases. In contrast to the relational algebra, an algebra for interactive multimedia presentations must operate on trees whose branches reflect different possible playouts of a family of presentations. The query language supports selection type operations for locating objects and presentation paths that are of interest to the user, join type operations for combining presentations from multiple databases into a single presentation, and finally set theoretic operations for comparing different databases. The algebra operations can be used to locate presentations with specific properties and also for creating new presentations by borrowing different components from existing ones. We prove a host of equivalence results for queries in this algebra which may be used to build query optimizers for interactive presentation databases.

1 Introduction

Though there are now literally hundreds of thousands of interactive multimedia presentations all over the world (ranging from PowerPoint presentations to Macromedia Director and Asymetrix Toolbook presentations), the concept of an *interactive multimedia presentation database* has not been adequately explored, though an important preliminary investigation (which did not account for interaction) was explored by Özsoyoglu et.al. [11]. In this paper, we investigate the data management support needed to query interactive multimedia presentations (IMPs for short) from possibly heterogeneous repositories and to create modified presentations on the fly from these repositories. An interactive presentation combines objects that are bound by special presentation constraints within a single node into a special tree-like structure that shows how users can navigate from one presentation to another. Such a structure exists in almost all static documents that control the playback of objects with items such as radio buttons, image maps, hyperlinks and text fields that load a new presentation. For example, a PowerPoint presentation is usually converted to such a structure. Other examples of interactive presentations are any list of linked document pages that contain dynamic components (dynamic HTML, javascript, etc.) that involve text, image, video and audio objects organized into component visual frames and presentation schedules. Multimedia authoring should not only involve the "exact" specification of how the objects come together in the playout phase, but it should also support tools for the specification of what the content of nodes are and what the result of different interactions are.

This paper is an attempt to formalize the querying of interactive multimedia presentations. It builds on top of previous work [5] in which we have developed the concept of an interactive multimedia presentation or IMP for short, and implemented it. In this paper, we make the following unique contributions:

- We give the formal definition of an interactive presentation database using generic notions of objects, multimedia presentation nodes and interactions. This description, as used in previous work, allows the creation of index structures for presentations developed under specialized programs, and in many cases presented using specialized programs.
- We define a *multimedia presentation algebra* (MPA) that contains generalizations of select, project and join operations in the relational algebra. The operators allow the querying of interactive presentation databases based on the contents of individual nodes as well as their structure induced by different interactions. In particular we develop specialized operators for
  - unary operations such as *select* and *project* that

allow queries to focus attention on certain objects and tree structures, and
- binary operations such as *merge, join, path-union, path-intersection* and *path-difference* that extend algebraic join operation to tree structures, allowing the queries to construct new presentations by combining the existing presentations in a multitude of ways.

The algebra allows users to query databases based on their content and structure. At the same time, it allows users to describe new presentations based on an existing database. Hence, the language can also be used as an authoring tool for describing higher level IMP views that facilitate the creation, modification and re-use of component presentations independent of the views presented to various users.

- We develop a host of equivalence results that form the basis of rewrite rules for query optimization in interactive presentation databases, allowing the efficient creation of different presentations at query execution time.

2 Interactive Multimedia Presentation Databases

In this section, we give the formal definition of interactive multimedia presentations (IMP). Intuitively, an IMP is a collection of presentation nodes where each node is a single multimedia presentation involving multiple objects, visible and/or audible. An interaction models user navigation from one node to another using any type of input, such as the push of a button, a voice command or entering text in a search field. Below, we define each of these concepts formally.

Definition 2.1 [Presentation Node] A presentation node N is a pair $\langle CONT_N, INT_N \rangle$ where $CONT_N$ is the content of the node and INT_N is the set of all possible interactions on the contents of node N. The contents $CONT_N$ of a node is a triple $CONT_N = \langle O_N, SC_N, TC_N \rangle$ where O_N is a set of objects that contain both presentation and non-presentation objects, SC_N and TC_N are sets of spatial and temporal constraints on presentation objects in O_N.

An interaction I in INT_N is identified in the system with a unique identifier I_{id}. Each interaction consists of a 5-tuple of the form $\langle I_{id}, A_O, A_{SC}, A_{TC}, A_{int} \rangle$ where:

- I_{id} is a unique label given to the interaction,
- A_O is a set of objects actions of the form $o+$ or $o-$,
- A_{SC} is a set of spatial constraint actions of the form $sc+$ or $sc-$,
- A_{TC} is a set of temporal constraint actions of the form $tc+$ or $tc-$, and
- A_{int} is a set of interaction actions of the form $i+$ or $i-$ for some interaction identifier i.

In particular, we use the notation A_O+ to refer to the subset of A_O that contains all positive atoms, and A_O- to the subset with all negative atoms. A similar notation is adopted for other types of action atoms.

In the above definition, positive (negative) atoms are atoms that the interaction adds/removes from node N.

A *presentation object* o is any multimedia object (relational table, text document, image, video segment, audio segment, web page, etc.) with a unique identifier id and a known presentation method (or viewer/player). An object may have a type, many attributes and methods that define relations over objects. We use the general convention of $o.a$ to refer to the value of attribute a of object o. Objects can have sub-objects as a specific type of attribute. A non-presentation object is one that does not have a presentation method.

Spatial constraints describe where objects should appear in the presentation. For each such object o, we use four distinct methods (called spatial constraint terms), $ulc(o), llc(o), urc(o), lrc(o)$ to refer to the coordinates of the upper and lower left, upper and lower right corners of the object in the presentation. A *spatial constraint* is then an expression of the form $sct_1 - sct_2 \langle op \rangle c$ where sct_1 and sct_2 are spatial constraint terms, $\langle op \rangle \in \{=, \leq, <, >, \geq\}$ and c is an integer. Temporal constraints are defined analogously using the temporal terms $st(o)$ and $et(o)$ that refer to the start and end times of the presentation of object o. *Throughout this paper, all spatial and temporal constraints considered are assumed to be of the form $x - y \leq c$. Such constraints are usually called* **difference constraints.** A *presentation schedule* associated with a set TC_N of temporal constraints is any solution of TC_N. In other words, the system can choose any presentation schedule as long as it is consistent with the set of constraints at that node.

Definition 2.2 [Result of an Interaction] Suppose $\langle \langle O_N, SC_N, TC_N \rangle, INT_N \rangle$ is a presentation node and $\langle I_{id}, A_O, A_{SC}, A_{TC}, A_{int} \rangle$ is an interaction in INT_N. The node $N' \equiv \langle \langle O'_N, SC'_N, TC'_N \rangle, INT'_N \rangle$ is said to be the result of interaction I_{id} on N iff

- $O'_N = (O_N \setminus A_O-) \cup A_O+$,
- $SC'_N = (SC_N \setminus A_{SC}-) \cup A_{SC}+$,
- $TC'_N = (TC_N \setminus A_{TC}-) \cup A_{TC}+$,
- $INT'_N = (INT_N \setminus A_{int}-) \cup A_{int}+$.

Definition 2.3 [Interactive Multimedia Presentation (IMP) Space] An interactive multimedia presentation space (or IMP_Space for short) is a tree in which each node $N = \langle CONT_N, INT_N \rangle$ is a presentation node

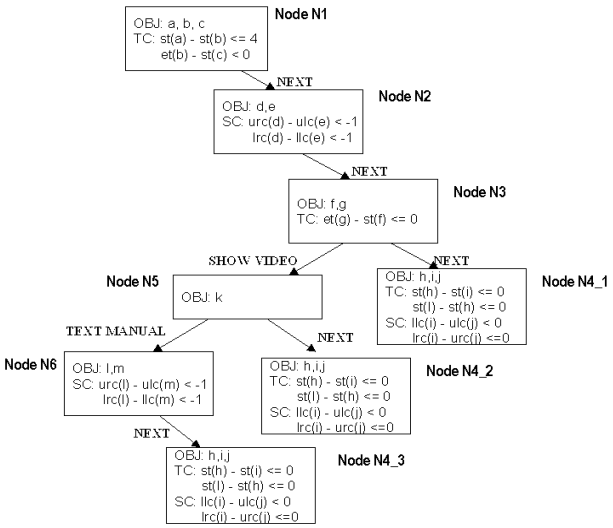

Figure 1: An Example Interactive Multimedia Presentation

and for each $i \in INT_N$, N has a single descendant node N' such that N' is the result of the interaction i on N. The root contains a special object *start* in its content with an interaction resulting in the first presentation, and all the leaf nodes contain only special *end* objects and no interactions.

Every path of non-zero length in an IMP is also an IMP. Each path in an IMP as well as the original IMP is assumed to have implicit root and leaf nodes with *start* and *end* objects.

Example 2.1 Before we go on any further, we introduce a small example that will be used throughout the paper to illustrate various operations on interactive multimedia presentations. Figure 1 contains a partial picture of an interactive multimedia presentation. The objects and corresponding spatial and temporal constraints are written in individual presentation nodes. A node is linked to another with the label of the interaction that leads to that node. The main interaction is called "next" which simply loads the next presentation in the logical sequence. Node N_4 can be reached in three different ways, each of which is represented as a separate path to ensure that the IMP has a tree structure. We have given each (virtual) copy of the node a different index to differentiate different paths in the IMP. This type of extensions of interactions may lead to potentially infinite interactive multimedia presentations. We are currently extending this algebra to graph structures to remedy this situation.

The following are desirable properties of interactive multimedia presentations (IMP):

- *Compact.* An IMP is compact if it does not contain any empty nodes, i.e. nodes with no objects.
- *Bound.* An IMP is bound if all the constraints in individual nodes involve objects located in that node.
- *Realizable.* An IMP is realizable if the constraints in the nodes are all satisfiable constraints.

An IMP is *valid* if it satisfies the realizability and boundness conditions. An IMP is *strongly valid* if in addition, it is compact. In the rest of the paper, we assume that all IMPs are valid. In general, compactness can be ensured by removing empty nodes and adjusting the interactions. Similarly, boundness can be ensured by *restricting* all constraints to the relevant object terms and unrealizable constraints can be *relaxed* by removing minimal conditions from the constraints. The restriction of a constraint set C to a set of objects O is given by a set of constraints C' that have identical solutions to C w.r.t. objects O. In [2], we develop algorithms for ensuring strong validity based on declarative notions of constraint restriction and relaxation. Due to space considerations, we indicate when MPA operations preserve validity and strong validity automatically. In other cases, we assume that the appropriate action can be taken to ensure these properties.

An *interactive presentation database* **IPD** is then a finite set of interactive presentations.

3 Selection Conditions in MPA

In relational algebra, selection expressions are built around attribute and relation names. MPA expressions are built around objects, nodes and paths. We use O for object variables, N for node variables, and $\wp$ for path variables.

3.1 Object Selection Conditions in MPA

Object selection conditions contain variables that range over different objects appearing in the nodes and denote a set of objects that satisfy the given condition. We assume that all media objects have an associated "type" that describes one or more attributes of the object. The set of types includes primitive types (e.g. `real, integer, string, avi, mpg, jpg,` etc.) and complex types constructed from the primitive types using array and record constructions. In order to define *object selection conditions*, we assume the existence of a set of variables ranging over the base types. Variables over complex types are inductively defined as follows. Suppose O is a variable over objects of type τ. Then:

- If $\tau = \text{ARRAY}[1, \ldots, K]$ of τ' then $O[i]$ is a variable of type τ' for $1 \leq i \leq K$.
- If $\tau = \text{RECORD} f_1 : \tau_1, \ldots, f_n : \tau_n \text{END RECORD}$ then $O.f_i$ is a variable of type τ_i.

- If o is an object variable, then $st(O), et(O), llc(O), lrc(O), ulc(O), urc(O)$ are variables of type real (corresponding to the start and end times, lower/upper left/right corners of an object in a presentation).

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 both *atomic selection conditions*. 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 an *atomic selection condition*, where $\Re$ is either $\{<, \leq, >, \geq\}$.

Weak Selection conditions may now be inductively defined as follows: (i) every atomic selection condition is a weak selection condition, (ii) if $\mathsf{woc}_1, \mathsf{woc}_2$ are weak object selection conditions, then so are $(\mathsf{woc}_1 \wedge \mathsf{woc}_2)$ and $(\mathsf{woc}_1 \vee \mathsf{woc}_2)$. An *object selection condition*, denoted by oc, is a weak object selection condition containing exactly one object variable symbol in it. For simplicity, we assume that the special objects *start* and *end* satisfy all (weak) object selection conditions.

Example 3.1 The object selection condition $O.media =$ "$video$" denotes all video objects. Similarly, $et(O) \leq 10$ denotes all objects that are played within the first 10 minutes of the presentation. Other examples of atomic selection conditions are: $O.creator =$"Jeff Ullman", $O.owner\_org =$ "Stanford University", and $O.creation\_date \leq (1, 1, 98)$. The last condition refers to objects created before Jan. 1., 1998 where $\leq$ is the usual "before" operation on dates. Similarly, the condition $(O.media\_type =$ "$video$") $\wedge$ $(et(O) \leq 10)$ constrains the selection condition to all video objects that are played within the first 10 minutes.

3.2 Node Selection Conditions in MPA

Now, we extend the definition of object selection conditions to *node selection conditions*. Intuitively, an object selection condition describes a set of objects regardless of which node they may appear in. Node selection conditions describe nodes based on their contents.

In addition, we also add constraints that describe the relative position of nodes in an IMP. A *complete path* in an IMP is any path from the root of the IMP to a leaf. If $N_1, \ldots, N_k$ is any complete path of an IMP, and $1 \leq i \leq j \leq k$, then $N_i, \ldots, N_j$ is called a *path* in that IMP. If N_1, N_2 are nodes in a path of an IMP, 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 IMP (assuming the root to be at level 0). If two nodes do not lie on a linear path, then the $\ominus$ is undefined for these nodes. For example, in Figure 1, $N2 \ominus N5 = 2$, $N4_2 \ominus N1 = -4$, and $N6 \ominus N4_1$ is undefined.

A *weak node selection condition* is defined as follows:

- If O is an object variable and o is an object-id, 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 $\mathsf{wnc} \wedge \mathsf{woc}$ is a weak node selection condition.
- If wnc_1 and wnc_2 are weak node selection conditions then so are $\mathsf{wnc}_1 \vee \mathsf{wnc}_2$, and $\mathsf{wnc}_1 \wedge \mathsf{wnc}_2$.

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

Example 3.2 We will expand the object selection conditions discussed above. The following expression
$(O_1 \in N) \wedge (O_1.media ='' text'') \wedge (O_2 \in N) \wedge$
$(O_2.media ='' video'') \wedge (O_1.title = O_2.title)$
finds all nodes that contain both a text and video object on the same subject, while the node selection condition
$(O \in N) \wedge (O.title ='' XYZ'') \wedge$
$(O.media ='' text'') \vee (O.media ='' video'')]$
finds all nodes that contain either a text or a video object on the subject "XYZ".

Recall the example IMP given in Example 2.1. Suppose in this example, only object k is of media type video, and objects a, b, f, g, h, i, l, m are all of type text. Of these, k, a, f, g, l all contain the title "XYZ". In this case, no nodes in this presentation satisfy the first condition. However, nodes 1,3,5, and 6 all satisfy the second node selection condition.

3.3 Path Selection Conditions in MPA

Finally, we describe *path selection conditions*. Path selection conditions denote linear paths in IMPs. We use the notation $\mathsf{paths}(\mathcal{I})$ to denote the set of all *complete* paths in an IMP $\mathcal{I}$ and $\mathsf{paths}(\mathsf{ip}\mathcal{DB})$ to denote the set $\bigcup_{\mathcal{I} \in \mathsf{ip}\mathcal{DB}} \mathsf{paths}(\mathcal{I})$. Hence, the set $\mathsf{paths}(\mathcal{I})$ contains a set of IMPs with explicit root and leaf nodes, constructed from paths in $\mathcal{I}$ hence leaving the content of the nodes unchanged, except all nodes in $\mathsf{paths}(\mathcal{I})$ contain a single interaction that leads to its child node. A path variable $\wp$ for a given $\mathsf{ip}\mathcal{DB}$ ranges over all complete and partial paths over $\mathsf{paths}(\mathsf{ip}\mathcal{DB})$.

- If N is a node variable, then $N \in \wp$ is a weak path selection condition.
- If wnc is a weak node selection condition and wpc is a weak path selection condition then, $(\mathsf{wpc} \wedge \mathsf{wnc})$ is a weak path selection condition.
- If $\mathsf{wpc}_1, \mathsf{wpc}_2$ are weak path selection conditions, then so are $\mathsf{wpc}_1 \wedge \mathsf{wpc}_2$ and $\mathsf{wpc}_1 \vee \mathsf{wpc}_2$.

A *path selection condition* is a weak path selection condition that contains exactly one path variable. The satisfaction of object and node conditions are defined in the obvious way. The path conditions are only satisfied by "minimal" paths. A path minimally satisfies a condition if it satisfies the condition, and none of its sub-paths does.

Example 3.3 Now, suppose we want to find a path where a node containing a text presentation is immediately followed by a video presentation on the same subject. We can express this using a path selection condition as follows:

$$(N_1 \in \wp) \wedge (N_2 \in \wp) \wedge (O_1 \in N_1) \wedge (O_2 \in N_2) \wedge (O_1.media ='' text'') \wedge (O_2.media ='' video'') \wedge (O_1.title = O_2.title) \wedge (N_2 \ominus N_1 = 1).$$

The expression $(N_1 \in \wp) \wedge (N_2 \in \wp) \wedge (N_2 \ominus N_1 = 4)$ selects all directed paths of length 4. If we want to find paths that start with a node containing a text object with title "XYZ" followed in three or less nodes by a video with the same title, then we can write the path selection condition:

$$(N_1 \in \wp) \wedge (N_2 \in \wp) \wedge (O_1 \in N_1) \wedge (O_2 \in N_2) \wedge (O_1.title ='' XYZ'') \wedge (O_1.media ='' text'') \wedge (O_2.media ='' video'') \wedge (O_1.title = O_2.title) \wedge (N_2 \ominus N_1 \leq 3) \wedge (N_2 \ominus N_1 > 0).$$

In this example, even though two distinct node and object variables are used to form a weak object and node selection condition, there is a single path variable, namely $\wp$. Hence, this expression is a path selection condition.

Recall the example IMP given in Example 2.1. Suppose that objects a, f, g, l are text objects and k is a video object that satisfy the above object conditions. Hence, we can set N_1 above to either one of the nodes 1,3, and 6. We can set N_2 to node 5 above. There are three possible paths for $\wp$ above, path $1 \ldots 5$, $3 \ldots 5$ and $6 \ldots 5$. We find out that $(N_2 = 5) \ominus (N_2 = 1) = 3$, $(N_2 = 5) \ominus (N_1 = 3) = 1$, $(N_2 = 5) \ominus (N_2 = 6) = -1$. Finally, we conclude that only the path $1 \ldots 5$ satisfies all the conditions in the above expression.

4 MPA Operations

Figure 2 gives the complete list of the operations in the algebra together with their syntax and intended interpretation.

4.1 Single Presentation Operations

These operations apply to single presentations in a given database. They select the desired content from all nodes or project out undesired paths from tree. The two operations are SELECT (σ) and PROJECT (π).

4.1.1 Selection

The selection operator is similar to the selection in relational algebra. It selects objects from nodes that satisfy an object selection condition, and discards the rest. Hence, expression $\sigma_O[\mathsf{oc}](\mathsf{ip}\mathcal{DB})$ is a selection operation with respect to object selection condition oc on variable O and the database $\mathsf{ip}\mathcal{DB}$.

Example 4.1 We can use any of the object selection conditions given earlier as a basis of a selection operation. Hence, the expression

$\sigma_O[(O.creator =$ "Jeff Ullman$'') \vee$
$(O.media =$ "$video'' \wedge et(O) \leq 10)](\mathsf{ip}\mathcal{DB})$.

returns all IMPs that contain only objects that are created by "Jeff Ullman" or any video object that is played within the first 10 minutes of the presentation. In other words, it restricts all nodes in all IMPs in the database to the objects mentioned in the selection operation.

We first define the effect of a selection operation to a single presentation node, and then give the general definition of selection on IMPs.

Definition 4.1 [Atomic Selection on a Single Node] Suppose $\langle CONT_N, INT_N \rangle$ is a node, where $CONT_N = \langle O_N, SC_N, TC_N \rangle$. Suppose oc is a object selection condition involving object variable O. The result $\sigma_O[\mathsf{oc}](N)$ of applying the selection is defined as a node $\langle CONT_{N'}, INT_{N'} \rangle$ where $CONT_{N'} = \langle O_{N'}, SC_{N'}, TC_{N'} \rangle$ is such that:

1. $O_{N'}$ is the set of all objects in O_N satisfying condition oc (i.e. for all $o \in O_{N'}$, we have that $o \models_N \mathsf{oc}$), and
2. $TC_{N'}$ is a set of temporal difference constraints such that the $TC_{N'}$ is a restriction of TC_N to the objects in $O_{N'}$, and
3. $SC_{N'}$ is a set of spatial difference constraints such that the $SC_{N'}$ is a restriction of SC_N to the objects in $O_{N'}$.
4. $INT_{N'}$ is equivalent to INT_N, except for all interactions $i \equiv \langle i_{id}, A_O, A_{SC}, A_{TC}, A_{int} \rangle \in INT_N$, then both A^+_{SC} and A^+_{SC} are restricted to objects in $O_{N'}$.

Furthermore, if $N_p = \langle \langle O_{N_p}, SC_{N_p}, TC_{N_p} \rangle, INT_{N_p} \rangle$ is the parent of N and $i \equiv \langle i_{id}, A_O, A_{SC}, A_{TC}, A_{int} \rangle \in INT_{N_p}$, is the interaction resulting in N, we replace i with the following $\langle i_{id}, A^\sigma_O, A^\sigma_{SC}, A^\sigma_{TC}, A_{int} \rangle$, where $A^\sigma_O = A_O \setminus \{ o^+ \mid o \in (O_{N_p} \setminus O_{N'}) \}$, $A^\sigma_{SC} = A^{\sigma+}_{SC} \cup A^{\sigma-}_{SC}$, where $A^{\sigma-}_{SC} = A^-_{SC}$, and $A^{\sigma+}_{SC}$ is the restriction of A^+_{SC} to the objects in $O_{N'}$.

Analogously, $A^\sigma_{TC} = A^{\sigma+}_{TC} \cup A^{\sigma-}_{TC}$, where $A^{\sigma-}_{TC} = A^-_{TC}$, and $A^{\sigma+}_{TC}$ is the restriction of A^+_{TC} to the objects in $O_{N'}$.

Operator	Syntax	Interpretation
Select	$\sigma_O[\mathsf{oc}](\mathsf{ip}\mathcal{DB})$	Restrict all nodes to objects O that satisfy condition oc
Project	$\pi_\wp[\mathsf{pc}](\mathsf{ip}\mathcal{DB})$	Remove all nodes that do not lie on a path $\wp$ that satisfies condition pc
Merge	$\mathsf{ip}\mathcal{DB}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathsf{ip}\mathcal{DB}'[\mathsf{wpc}_2]$	For all pairs of paths $\wp_1, \wp_2$ that satisfy conditions $\mathsf{wpc}_1, \mathsf{wpc}_2$ merge the contents of all nodes on $\wp_2$ into nodes on $\wp_1$
Merge Closure	$\mathsf{ip}\mathcal{DB}[\mathsf{wpc}_1] \overset{*}{\lhd}^{\mathsf{cm}}_{\wp_1:\wp_2} \mathsf{ip}\mathcal{DB}'[\mathsf{wpc}_2]$	Merge the contents of nodes on $\wp_2$ into nodes on $\wp_1$ for all possible paths $\wp_1, \wp_2$ that satisfy conditions $\mathsf{wpc}_1, \mathsf{wpc}_2$
Join	$\mathsf{ip}\mathcal{DB}[\mathsf{wnc}] \bowtie_{N:\wp} \mathsf{ip}\mathcal{DB}'[\mathsf{wpc}]$	For all pairs of node N, path $\wp$ that satisfy conditions $\mathsf{wnc}, \mathsf{wpc}$, insert $\wp$ after node N
Join Closure	$\mathsf{ip}\mathcal{DB}[\mathsf{wnc}] \overset{*}{\bowtie}_{N:\wp} \mathsf{ip}\mathcal{DB}'[\mathsf{wpc}]$	Paste all paths $\wp$ from $\mathsf{ip}\mathcal{DB}'$ that satisfy the condition wpc after all nodes N in $\mathsf{ip}\mathcal{DB}$ that satisfy wnc
Path-Union	$\mathsf{ip}\mathcal{DB} + \mathsf{ip}\mathcal{DB}'$	Combine all possible paths in $\mathsf{ip}\mathcal{DB}$ and $\mathsf{ip}\mathcal{DB}'$ into a single IMP
Path-Intersection	$\mathsf{ip}\mathcal{DB} * \mathsf{ip}\mathcal{DB}'$	Construct an IMP that contains only the paths common to all IMPs in $\mathsf{ip}\mathcal{DB} \cup \mathsf{ip}\mathcal{DB}'$
Path-Difference	$\mathsf{ip}\mathcal{DB} \setminus \mathsf{ip}\mathcal{DB}'$	Remove from $\mathsf{ip}\mathcal{DB}$ all paths that appear in $\mathsf{ip}\mathcal{DB}'$
Set-Union	$\mathsf{ip}\mathcal{DB} \cup \mathsf{ip}\mathcal{DB}'$	Create a new database containing all presentations from either database
Set-Intersection	$\mathsf{ip}\mathcal{DB} \cap \mathsf{ip}\mathcal{DB}'$	Create a new database containing all presentations that are in both databases
Set-Difference	$\mathsf{ip}\mathcal{DB} - \mathsf{ip}\mathcal{DB}'$	Create a new database containing all presentations that are only in $\mathsf{ip}\mathcal{DB}$

Figure 2: Complete List of Multimedia Presentation Algebra Operations

Hence, the selection operator simply selects the objects satisfying the given condition, then simplifies the spatial and temporal constraints for that node to preserve its validity. We are now ready to define the *Selection* operator on IDBs.

Definition 4.2 [Selection on an IPD] Suppose $\mathcal{I}$ is an interactive multimedia presentation and oc is an object selection condition. Then *the result of selection operator* oc *on* $\mathcal{I}$ is the IMP $\mathcal{I}'$ obtained by applying the node selection operation $\sigma_O[\mathsf{oc}](N)$ to each node N.

If $\mathsf{ip}\mathcal{DB}$ is an interactive presentation database, then $\sigma_O[\mathsf{oc}](\mathsf{ip}\mathcal{DB}) = \{\sigma_O[\mathsf{oc}](\mathcal{I}) \mid \mathcal{I} \in \mathsf{ip}\mathcal{DB}\}$.

Example 4.2 Consider the simple example $\mathsf{ip}\mathcal{DB}$ shown in Figure 1. Suppose our database contains only this presentation. Suppose now given a selection operation $\sigma_O[\mathsf{oc}](\mathsf{ip}\mathcal{DB})$, only objects a, b, c, f, g, h, m satisfy the selection condition. Then, as a result of the selection query, nodes 2 and 5 become completely empty. Nodes N1 and N3 are unchanged, node N4 is restricted to object h. This means, all spatial and temporal constraints for this node are eliminated, and node N6 is restricted to only object m.

Note that the IMP $\mathcal{I}$ and $\sigma_O[\mathsf{oc}](\mathcal{I})$ are isomorphic, i.e., they have the same tree structure. The actual computation of the selection operation depends on the computation of the restriction operation on the constraints of individual nodes. We show in [2] that it is always possible to find a "minimal" restriction of the given constraints.

Theorem 4.1 If $\mathsf{ip}\mathcal{DB}$ is a set of valid interactive presentations, then so is $\sigma_O[\mathsf{oc}](\mathsf{ip}\mathcal{DB})$.

We note however that the select operation does not necessarily preserve strong validity. To ensure this, nodes with empty content must be removed from all IMPs. Finally, we show below that selection is commutative.

Theorem 4.2 $\sigma_{O_1}[\mathsf{oc}_1](\sigma_{O_2}[\mathsf{oc}_2](\mathsf{ip}\mathcal{DB})) = \sigma_{O_2}[\mathsf{oc}_2](\sigma_{O_1}[\mathsf{oc}_1](\mathsf{ip}\mathcal{DB}))$.

Corollary 4.1 $\sigma_{O_1}[\mathsf{oc}_1 \wedge \mathsf{oc}_2](\mathsf{ip}\mathcal{DB})) = \sigma_{O_1}[\mathsf{oc}_1](\sigma_{O_1}[\mathsf{oc}_2](\mathsf{ip}\mathcal{DB})) = \sigma_{O_1}[\mathsf{oc}_2](\sigma_{O_1}[\mathsf{oc}_1](\mathsf{ip}\mathcal{DB}))$.

4.1.2 Projection

Selections as defined above, only consider the objects within one node, but do not consider the relationship between nodes. Projection finds specific paths that satisfy a path selection condition and removes all nodes that do not lie on one of the selected paths.

Definition 4.3 [Projection Operator] Suppose $\mathcal{I}$ is an interactive presentation, and pc is a path selection condition involving path variable $\wp$. The projection of $\mathcal{I}$ w.r.t. pc, denoted $\pi_\wp[\mathsf{pc}](\mathcal{I})$, is the IMP obtained by eliminating from $\mathcal{I}$, all nodes that do not lie on a path $\wp$ that satisfies pc.

The projection of an interactive presentation database, $\mathsf{ip}\mathcal{DB}$ w.r.t. a path selection condition pc, denoted by $\pi_\wp[\mathsf{pc}](\mathsf{ip}\mathcal{DB}) = \{\pi_\wp[\mathsf{pc}](\mathcal{I}) \mid \mathcal{I} \in \mathsf{ip}\mathcal{DB}\}$.

Example 4.3 Consider the database containing the IMP shown in Figure 1. Suppose we write the following:

$\pi_\wp[(a \in N_1) \wedge [(h \in N_2) \vee (k \in N_2) \wedge (N_1 \ominus N_2 \leq 4) \wedge (N_1 \in \wp) \wedge (N_2 \in \wp)](\mathsf{ip}\mathcal{DB})$

This projection operation chooses all paths of length 4 or less that extend from a node containing object a to

a node containing object h. If we apply this to the tree shown in Figure 1, we end up removing nodes $N4_1, N4_3$ and $N6$.

Theorem 4.3 The projection operator preserves both validity and strong validity.

1. $\pi_{\wp}[\mathsf{pc}](\mathcal{I})$ is valid whenever $\mathcal{I}$ is valid.
2. $\pi_{\wp}[\mathsf{pc}](\mathcal{I})$ is strongly valid whenever $\mathcal{I}$ is strongly valid.

However, projection and selection are not commutative, i.e. $\pi_{\wp}[\mathsf{pc}](\sigma_O[\mathsf{oc}](\mathsf{ip}\mathcal{DB})) = \sigma_O[\mathsf{oc}](\pi_{\wp}[\mathsf{pc}](\mathsf{ip}\mathcal{DB}))$ is not necessarily true. This is no surprise as projection cannot be pushed down a selection even in the relational algebra if the selection involves attributes that are removed by a projection operation. As a counter-example in the MPA case suppose we have a tree $\mathcal{I}$ with 2 nodes, N_A and N_B, with $O_A = \{a, b, c\}$ and $O_B = \{d, c\}$, with the following attributes, a.title = "XX", a.media = "video", b.title = "XX", b.media = "text", d.title = "XX", d.title = "text", c.title = "XX", c.media = "video". Consider the path condition pc given below:
$(N_1 \in \wp) \wedge (N_2 \in \wp) \wedge (O_1 \in N_1) \wedge (N_2 \ominus N_1 > 0) \wedge$
$(O_1.title = "XX") \wedge (O_1.media = "video") \wedge$
$(O_2 \in N_2) \wedge (O_2.title = "XX") \wedge (O_2.media = "text")$.

The projection of $\mathcal{I}$ w.r.t. pc is exactly $\mathcal{I}$. Consider the selection condition $\mathsf{oc} \equiv O.media = "text"$. Then $\sigma_O[\mathsf{oc}]\{\mathcal{I}\} = \sigma_O[\mathsf{oc}](\pi_{\wp}[\mathsf{pc}]\{\mathcal{I}\})$ contains a single IMP with two nodes whose content is given by, $O'_A = \{b\}$, and $O'_b = \{d\}$. However, $\pi_{\wp}[\mathsf{pc}](\sigma_O[\mathsf{oc}]\{\mathcal{I}\})$ is an empty IMP. Below, we prove a weaker form of commutativity.

Theorem 4.4 Let $\mathsf{ip}\mathcal{DB}$ be a presentation databases, $O_1, \ldots, O_m$ be all the object variables occurring in pc, and O be the object variable in oc. Assume pc and oc do not share any variables. Selection can be pushed down a projection operation, i.e. $\sigma_O[\mathsf{oc}](\pi_{\wp}[\mathsf{pc}]\mathsf{ip}\mathcal{DB}) = \pi_{\wp}[\mathsf{pc}](\sigma_O[\mathsf{oc}]\mathsf{ip}\mathcal{DB})$ if the condition $\mathsf{pc} \wedge (\bigvee_{i=1}^{m} not(\mathsf{oc}/_{O=O_i}))$ is unsatisfiable for all possible interactive presentation databases. (This condition guarantees that every object variable that makes pc true, also satisfies $\bigwedge_{i=1}^{m}(\mathsf{oc}/O = O_i)$).

4.2 Binary Presentation Operations

As discussed earlier, these operations range over pairs of presentations from two different databases and compute the "join" of two IMPs with respect to different criteria. The intuition for each operator is given below.

MERGE ($\lhd$). The merge operation affects the contents of individual nodes. The input to the merge operation is two different IMPs. The output has the same structure as the first input presentation except for the contents of the nodes. The nodes have been expanded through the addition of new objects and constraints drawn from the second input presentation. For example, we might want to add some text right next to the video presentation of a software program. Then the presentation of the text object is inserted into the node that contains the video presentation with adding the object together with necessary spatial and temporal constraints.

JOIN ($\bowtie$). The join operation takes two interactive presentations as input and returns a single interactive presentation as output. In this case, two presentations are joined by inserting nodes from each one into a new tree structure. Again, the main difference between the merge operation and the join operation is that, the contents of the nodes remain the same, only the overall structure of the input presentation changes. For example, we might want to insert a video presentation of a new product just before the text based presentations that describe the product.

PATH-UNION ($+$): The union of two presentations is a new presentation whose set of paths coincides with the union of the set of paths of the two presentations. Path-Intersection ($*$) and Path-Difference ($-$) are defined analogously.

4.2.1 Merge

We are now ready to define a *merge* operation, $\lhd$. We will first define merge operations on two IMPs $\mathcal{I}, \mathcal{I}'$. The merge operation is defined with respect to a path selection condition that selects paths of same length from two different IMPs. The merge combines corresponding nodes on these two paths. The merge is an asymmetric operation, hence the merge of $\mathcal{I}$ and $\mathcal{I}'$ (denoted by $\mathcal{I} \lhd \mathcal{I}'$) will return as output, an IMP that has the same structure as $\mathcal{I}$. However, the contents of the nodes in $\mathcal{I}$ are "expanded" by the addition of new objects, and new constraints from $\mathcal{I}'$.

Definition 4.4 [Constraint Merge Functions] A *constraint merge function* cm takes as input, two sets of constraints, and returns as output, a new set of constraints such that $\mathsf{cm}(C_1, C_2) \supseteq C_1 \cup C_2$.

For example, cm may be used to specify that when merging the constraints of two nodes, then the value of all variables in the second node must be greater than or equal to the value of all variables in the first. This constraint merge function, called the *sequential constraint merge function* cm_s may be defined as $\mathsf{cm}_s(C_1, C_2) = C_1 \cup C_2 \cup \{x - y \leq 0 \mid x$ is a variable in C_1 and y is a variable in $C_2\}$.

Other constraint merge functions may also be defined. For example, we may "slide" a presentation in time and/or in space while merging it with another. Examples of such functions are "left" and "bottom" for space, "before" and "after" for time.

Definition 4.5 [Merging Two Nodes] Suppose n_1 and n_2 are two nodes. Suppose also that n_1 is a node in an IMP $\mathcal{I}$. The *merge of nodes* n_1 *and* n_2 *w.r.t. constraint merge function* $\mathsf{cm}_{s,t}$ in $\mathcal{I}$, denoted $\mu_{\mathsf{cm}_{s,t}}(\mathcal{I}, \mathsf{n}_1, \mathsf{n}_2)$ is the IMP that is identical to $\mathcal{I}$ except that n_1 is replaced with the node

$$\mathsf{n} = \langle \langle O_{\mathsf{n}_1} \cup O_{\mathsf{n}_2}, \mathsf{cm}_s(SC_1, SC_2), \mathsf{cm}_t(TC_1, TC_2) \rangle, Int_{\mathsf{n}} \rangle.$$

where $Int_{\mathsf{n}} = \{ \langle i_{id}, A^{\mu}_O, A^{\mu}_{SC}, A^{\mu}_{TC}, A^{\mu}_{int} \rangle \mid \langle i_{id}, A_O, A_{SC}, A_{TC}, A_{int} \rangle \in Int_{\mathsf{n}_1} \}$, $A^{\mu}_O = A_O \cup \{ o^- \mid o \in (O_{\mathsf{n}_2} \setminus O_{\mathsf{n}_1}) \}$, $A^{\mu}_{SC} = A_{SC} \cup \{ c- \mid c \in \mathsf{cm}_s(SC_1, SC_2) - SC_1 \}$, and $A^{\mu}_{TC} = A_{TC} \cup \{ c- \mid c \in \mathsf{cm}_t(TC_1, TC_2) - TC_1 \}$.

In addition, if n_p is the parent of node n then, if $i \equiv \langle i_{id}, A_O, A_{SC}, A_{TC}, A_{int} \rangle$ is the interaction on n_p that results in node n_1, then i is replaced with $\langle i_{id}, A_O \cup \{ o+ \mid o \in O_{\mathsf{n}_2} \}, A_{SC} \cup \{ c+ \mid c \in (\mathsf{cm}_s(SC_1, SC_2) \setminus SC_1) \}, A_{TC} \cup \{ c+ \mid c \in (\mathsf{cm}_t(TC_1, TC_2) \setminus TC_1) \}, A_{int} \rangle$.

Definition 4.6 [Merging Two Paths] Suppose $\mathsf{p}_1 \equiv \mathsf{n}_{1,1} \ldots \mathsf{n}_{1,k}$, and $\mathsf{p}_2 \equiv \mathsf{n}_{2,1} \ldots \mathsf{n}_{2,k}$ are two paths from $\mathcal{I}$ and $\mathcal{I}'$ respectively. Then, the merge of p_1 and p_2, denoted by $\mu^{+}_{\mathsf{cm}_{s,t}}(\mathcal{I}, \mathsf{p}_1, \mathsf{p}_2)$ is given by the following expression:

$$\mu^{+}_{\mathsf{cm}_{s,t}}(\mathcal{I}, \mathsf{p}_1, \mathsf{p}_2) = \mu_{\mathsf{cm}}(\ldots \mu_{\mathsf{cm}}(\mu_{cm}(I, \mathsf{n}_{1,1}, \mathsf{n}_{2,1}), \mathsf{n}_{1,2}, \mathsf{n}_{2,2}) \ldots \mathsf{n}_{1,k}, \mathsf{n}_{2,k}).$$

When extending the definition of merge to apply to two IMPs $\mathcal{I}, \mathcal{I}'$, we need to somehow specify which nodes in $\mathcal{I}$ get merged with which nodes in $\mathcal{I}'$. To do this, we specify two paths of equal length from the two IMPs. The nodes of the two paths will then be merged in the same sequence as they appear on the path. If more than one path satisfies the path condition, then a merged IMP is computed for each pair matching paths. The case of merging single nodes is a special case of this operation that uses paths involving a single node.

Definition 4.7 [Merging Two IMPs] Let $\mathcal{I}$ and $\mathcal{I}'$ be two IMPs, wpc_1 and wpc_2 are weak path selection conditions that involve path variables $\wp_1$ and $\wp_2$ respectively, then the expression $\mathcal{I}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}'[\mathsf{wpc}_2]$ is a merge expression. Then, the result of the merge operation is given by:

$$\mathcal{I}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}'[\mathsf{wpc}_2] = \{ \mu^{+}_{\mathsf{cm}_{s,t}}(\mathcal{I}, \mathsf{p}_1, \mathsf{p}_2) \mid \mathsf{wpc}_1/\{\wp_1 = \mathsf{p}_1\} \text{ and } \mathsf{wpc}_2/\{\wp_2 = \mathsf{p}_2\} \text{ are satisfiable} \}.$$

Definition 4.8 [Merging Two ip$\mathcal{DB}$s] Suppose ip$\mathcal{DB}$, ip$\mathcal{DB}'$ are two interactive presentation databases. Then the merge of ip$\mathcal{DB}$, ip$\mathcal{DB}'$ w.r.t. constraint merge function cm and a weak path selection condition is given by

$$\mathsf{ip}\mathcal{DB}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathsf{ip}\mathcal{DB}'[\mathsf{wpc}_2] = \bigcup \{ (\mathcal{I}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}'[\mathsf{wpc}_2]) \mid \mathcal{I} \in \mathsf{ip}\mathcal{DB}, \mathcal{I}' \in \mathsf{ip}\mathcal{DB}' \}.$$

Example 4.4 Consider the IMP given in Figure 1. Suppose for simplicity ip$\mathcal{DB}$ and ip$\mathcal{DB}'$ are two different databases that only contain this presentation. We write the following merge query:

$$\mathsf{ip}\mathcal{DB}[(a \in N_1) \wedge (d \in N_2) \wedge (N_1 \ominus N_2 \leq 1) \wedge (N_2 \ominus N_1 \leq -1) \wedge (N_1 \in \wp_1) \wedge (N_2 \wedge \wp_1)] \lhd^{left,after}_{\wp_1:\wp_2} \mathsf{ip}\mathcal{DB}'[(k \in N_3) \wedge (l \in N_4) \wedge (N_3 \ominus N_4 \leq 1) \wedge (N_4 \ominus N_3 \leq -1) \wedge (N_3 \in \wp_2) \wedge (N_4 \wedge \wp_2)]$$

This query asks for all presentations that result from the merge of a path of two nodes, containing objects a followed by d in ip$\mathcal{DB}$, and objects k and l in ip$\mathcal{DB}'$. Path $N1, N2$ satisfies the first path condition, and path $N5, N6$ satisfies the second condition. As a result, node $N5$ is merged into $N1$, and node $N6$ is merged into $N2$. In the result of this merge, new node $N2$ contains objects d, e, l, m, and the following spatial constraints:

$urc(d) - ulc(e) < -1$, $lrc(d) - llc(e) < -1$, $urc(l) - ulc(m) < -1$, $lrc(l) - llc(m) < -1$, $ulc(d) - urc(l) \leq 0$, $llc(d) - urc(l) \leq 0$, $ulc(d) - urc(m) \leq 0$, $llc(d) - urc(m) \leq 0$, $ulc(e) - urc(l) \leq 0$, $llc(e) - urc(l) \leq 0$, $ulc(e) - urc(m) \leq 0$, $llc(e) - urc(m) \leq 0$.

Unfortunately, merge has very limiting properties. The validity can only be proven for limited cases. For this reason, a widening operator must be used to relax the constraints to make sure they are satisfiable. Furthermore, selection and projection cannot be pushed down into merge. Since both operations may eliminate objects that would make the path expressions for merge to evaluate to true, the result of merge after a select/project is not necessarily equal to a merge before a select/project.

Theorem 4.5 If $\mathcal{I}, \mathcal{I}'$ are valid and do not share any objects, then $\mathcal{I}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}'[\mathsf{wpc}_2]$ is valid for any sequential constraint merge function cm.

The above merge operation is a binary operation, it takes two IMPs $\mathcal{I}, \mathcal{I}'$ and merges the contents of matching nodes from $\mathcal{I}'$ into $\mathcal{I}$ in a pairwise fashion, each time creating a different presentation. It is also possible to merge all paths satisfying the path condition from $\mathcal{I}'$ into all paths in $\mathcal{I}$. An example of such an operation is merging all pictures from a presentation into a single node in the other presentation. We now define this type of closure operator, which will be denoted by $\mathcal{I} \overset{*}{\lhd} \mathcal{I}'$.

Let $\mathcal{I}[\mathsf{wnc}_1] \overset{*}{\lhd}^{\mathsf{cm}}_{N_1:N_2} \mathcal{I}'[\mathsf{wnc}_2]$ be a closure merge operation. Let $\mathsf{p}_1, \ldots, \mathsf{p}_m$ and $\mathsf{p}'_1, \ldots, \mathsf{p}'_n$ be the complete list of paths from $\mathcal{I}$ and $\mathcal{I}'$ respectively that make wpc_1 and wpc_2 satisfiable. Then,

$\mathcal{I}[\mathsf{wpc}_1] \overset{*\,\mathsf{cm}}{\lhd}_{\wp_1:\wp_2} \mathcal{I}'[\mathsf{wpc}_2] = \mu^+_{\mathsf{cm}}(\ldots(\mu^+_{\mathsf{cm}}(\mu^+_{cm}(\mathcal{I}, \mathsf{p}_1, \mathsf{p}'_1), \mathsf{p}_1, \mathsf{p}'_2)\ldots), \mathsf{p}_1, \mathsf{p}'_n), \mathsf{p}_2, \mathsf{p}'_1), \mathsf{p}_2, \mathsf{p}'_2)\ldots), \mathsf{p}_2, \mathsf{p}'_n)\ldots, \mathsf{p}_m, \mathsf{p}'_1), \mathsf{p}_m, \mathsf{p}'_2)\ldots), \mathsf{p}_m, \mathsf{p}'_n))$.

Then, the closure merge of two presentation databases is computed by replacing the regular merge operation between two presentations with the closure operator. The validity result and all other results for merge also hold for merge closure. We finish this section with a weak commutativity result between merge and selection.

Theorem 4.6 Let $\mathsf{ip}\mathcal{DB}, \mathsf{ip}\mathcal{DB}'$ be presentation databases, $O_1, \ldots, O_m$ be all the object variables occurring in a weak path selection condition wpc_1, $O'_1, \ldots, O'_s$ be all the object variables occurring in weak path selection condition wpc_2, and O be the object variable in object selection condition oc.
Selection can be pushed down a merge operation, i.e. $\sigma_O[\mathsf{oc}](\mathsf{ip}\mathcal{DB}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathsf{ip}\mathcal{DB}'[\mathsf{wpc}_2]) =$
$(\sigma_O[\mathsf{oc}]\mathsf{ip}\mathcal{DB})[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} (\sigma_O[\mathsf{oc}]\mathsf{ip}\mathcal{DB}')[\mathsf{wpc}_2]$.
if both the conditions $\mathsf{wpc}_1 \wedge (\bigvee_{i=1}^{m} not(\mathsf{oc}/_{O=O_i}))$ and $\mathsf{wpc}_2 \wedge (\bigvee_{j=1}^{s} not(\mathsf{oc}/_{O=O'_j}))$ are unsatisfiable, for all possible interactive presentation databases.

4.2.2 Join

In this section, we will provide the formal definition of the "join" operator. Recall that given any two IMPs $\mathcal{I}, \mathcal{I}'$, the "merge" operator returns an IMP having the same structure as $\mathcal{I}$. In contrast, a join operation changes the structure of an IMP by adding new paths.

Definition 4.9 [Amalgam of IMPs] Suppose $\mathcal{I}, \mathcal{I}'$ are two IMPs, and n' is the root of $\mathcal{I}'$. The *amalgam of* $\mathcal{I}, \mathcal{I}'$ *on node* $\mathsf{n} \in \mathcal{I}$ *via action* a, denoted by $\mathcal{A}_{\mathsf{n},a}(\mathcal{I}, \mathcal{I}')$, is a tree that is rooted at the root of $\mathcal{I}$, that contains both subtrees $\mathcal{I}$ and $\mathcal{I}'$ except for the root of $\mathcal{I}'$, and the node n is modified as follows: If n is a leaf node containing an *end* object, then the parent of n is modified by replacing the interaction that results in n with an interaction a that results in the single child of n'. If n is an internal node, then a new interaction a is added to n that leads to the single child of n'.

Join is defined between a node and a path. Thus, joins are given by expressions of the form $\mathcal{I}[\mathsf{wnc}] \bowtie_{N:\wp} \mathcal{I}'[\mathsf{wpc}]$ where the paths $\wp$ from $\mathcal{I}'$ satisfying the weak path condition wpc will be amalgamated to the end of nodes N in $\mathcal{I}$ satisfying the weak node condition wnc.

Definition 4.10 [Joining Two IMPs] Suppose $\mathcal{I}, \mathcal{I}'$ are two IMPs, wnc is a weak node condition involving a node variable N, and wpc is a weak path condition involving a path variable $\wp$. Then, a join is an expression of the form $\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} \mathcal{I}'[\mathsf{wpc}]$. The result of a join operation between IMPs $\mathcal{I}$ and $\mathcal{I}'$ via action a is then defined as:

$\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} \mathcal{I}'[\mathsf{wpc}] = \{\mathcal{A}_{\mathsf{n},a}(\mathcal{I}, \mathsf{p}) \mid \mathsf{wnc}/\{N = \mathsf{n}\}$ and $\mathsf{wpc}/\{\wp = \mathsf{p}\}$ are satisfiable $\}$.

In the above definition, we made an implicit assumption that any path selected from an IMP is expanded when necessary with the special root and leaf nodes to explicitly put the selected path in the tree form. Hence, if $\mathsf{n}_1 \ldots \mathsf{n}_k$ is a proper path, i.e. that does not contain the root and leaf nodes, then this path is expanded to *start* $\mathsf{n}_1 \ldots \mathsf{n}_k$ *end*.

Definition 4.11 [Joining Two $\mathsf{ip}\mathcal{DB}$s] Let $\mathsf{ip}\mathcal{DB}, \mathsf{ip}\mathcal{DB}'$ be two interactive presentation databases. Then the join of $\mathsf{ip}\mathcal{DB}, \mathsf{ip}\mathcal{DB}'$ w.r.t. an interaction a is defined as $\mathsf{ip}\mathcal{DB}[\mathsf{wnc}] \bowtie^a_{N:\wp} \mathsf{ip}\mathcal{DB}'[\mathsf{wpc}] = \bigcup\{$
$(\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} \mathcal{I}'[\mathsf{wpc}]) \mid \mathcal{I} \in \mathsf{ip}\mathcal{DB}, \mathcal{I}' \in \mathsf{ip}\mathcal{DB}'\}$.

Example 4.5 Consider the IMP given in Figure 1. Suppose for simplicity $\mathsf{ip}\mathcal{DB}$ and $\mathsf{ip}\mathcal{DB}'$ are two different databases that only contain this presentation. We write the following join query:
$\mathsf{ip}\mathcal{DB}[(a \in N_1)] \bowtie_{N_1:\wp} \mathsf{ip}\mathcal{DB}'[(k \in N_2) \wedge (l \in N_3) \wedge (N_2 \in \wp) \wedge (N_3 \wedge \wp)]$
which joins all paths that contain two (not necessarily distinct) nodes, one containing object k and the other object l, after a node in $\mathsf{ip}\mathcal{DB}$ that contains object a. Since there is only one path that satisfies $\wp$, namely $N5, N6$, this path is pasted under node $N1$ as a result of this join operation.

Theorem 4.7 If $\mathcal{I}, \mathcal{I}'$ are valid, then $\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} \mathcal{I}'[\mathsf{wpc}]$ is valid. Furthermore, if $\mathcal{I}, \mathcal{I}'$ are strongly valid, then $\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} \mathcal{I}'[\mathsf{wpc}]$ is strongly valid.

However, as in the merge operation, selection and projections cannot be pushed down the join operation since this might eliminate certain paths that would make the join condition true.

The join closure operator is defined similar to the merge closure. The join closure of two IMPs result in a single IMP which contains all subtrees satisfying the path selection condition after all nodes satisfying the node selection condition. The join closure of two IMPs $\mathcal{I}, \mathcal{I}'$, denoted by $\mathcal{I}[\mathsf{wnc}] \overset{*\,a}{\bowtie}_{N:\wp} \mathcal{I}'[\mathsf{wpc}]$, for a weak node condition wnc involving a node term N and weak path condition wpc involving a path term $\wp$ is defined as follows. Let $\mathsf{n}_1, \ldots, \mathsf{n}_k$ be the list of all nodes in $\mathcal{I}$ that make wnc satisfiable when substituted for N, and let $\mathsf{p}_1, \ldots, \mathsf{p}_n$ be the list of all paths in $\mathcal{I}'$ that make wpc satisfiable when substituted for $\wp$. Then, we have the following:

$\mathcal{I}[\mathsf{wnc}] \overset{*\,a}{\bowtie}_{N:\wp} \mathcal{I}'[\mathsf{wpc}] = \mathcal{A}_{\mathsf{n}_k,a}(\ldots(\mathcal{A}_{\mathsf{n}_2,a}(\ldots(\mathcal{A}_{\mathsf{n}_2,a}(\mathcal{A}_{\mathsf{n}_1,a}(\ldots(\mathcal{A}_{\mathsf{n}_1,a}(\mathcal{A}_{\mathsf{n}_1,a}(\mathcal{I}, \mathsf{p}_1), \mathsf{p}_2)\ldots), \mathsf{p}_n), \mathsf{p}_1), \ldots), \mathsf{p}_n), \ldots), \mathsf{p}_n)$.

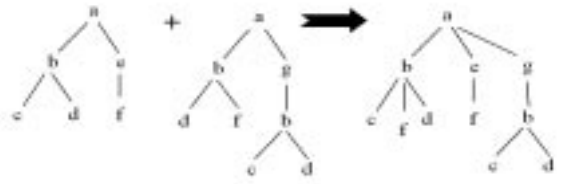

Figure 3: Path-Union Construction

4.2.3 Path Union, Intersection and Difference

When we consider an interactive multimedia presentation $\mathcal{I}$, each path in $\mathcal{I}$ reflects a possible playout of the presentation. The path union of two IMPs is a new IMP such that the set of paths in the new IMP is the union of the sets of paths in the two IMPs being "union"ed. Similar operations are defined with respect to paths that are common to all presentations in a database (path-intersection) and paths that are in one database, but not the other (path-difference).

Definition 4.12 [Combined Presentation] Let $\mathcal{I}_1, \ldots, \mathcal{I}_k$ be a set of IMPs that contain a single linear path. Then the combination of these, denoted by $\oplus\{\mathcal{I}_1, \ldots, \mathcal{I}_k\}$, is the minimal IMP $\mathcal{I}$ such that $\mathsf{paths}(\mathcal{I}) = \{\mathcal{I}_1, \ldots, \mathcal{I}_k\}$.

The minimality condition on $\mathcal{I}$ suggests that, there do not exist two nodes $\mathsf{n}_1 = \langle Cont, Int_1 \rangle$ and $\mathsf{n}_2 = \langle Cont, Int_2 \rangle$ in $\mathcal{I}$ with identical content that can be combined by letting $\mathsf{n}_1 = \langle Cont, Int_1 \cup Int_2 \rangle$ into a new tree structure $\mathcal{I}'$ with fewer nodes with the property $\mathsf{paths}(\mathcal{I}') = \{\mathcal{I}_1, \ldots, \mathcal{I}_k\}$.

Note that two trees representing IMPs are considered to be the same, irrespective of the order of the children. Let $\mathcal{P}$ be a set of (possibly infinite) *rooted paths*, i.e., paths whose starting node has $\langle \{start\}, \emptyset, \emptyset \rangle$ as its first component. Thus, the unique combination of $\mathcal{P}$, denoted $\oplus\mathcal{P}$ is the tree $\mathcal{I}$ defined as follows:

Consider the partition $\{\mathcal{P}_0, \mathcal{P}_1, \ldots\}$ of $\mathcal{P}$, where $\mathcal{P}_i = \{\mathsf{p} \in \mathcal{P} \mid i = length(\mathsf{p})\}$. Let $\mathcal{I}_0$ be the empty IMP, that is, the tree containing only one node, with content $\langle \{start\}, \emptyset, \emptyset \rangle$.

For every level $k \geq 0$, $\mathcal{I}_{k+1} = \oplus\mathcal{P}_{k+1}$ is obtained from $\mathcal{I}_k = \oplus\mathcal{P}_k$ by introducing, for every path p_i in $\mathcal{P}_{k+1}$, a unique copy of the $k+1$-st node - say $\mathsf{n}_{i,k+1}$ - as a child of the (unique) node $\mathsf{n}_{i,k}$ (which precedes n_{k_1} in p_i appearing in $\mathcal{I}_k$. If the partition is finite, say $\{\mathcal{P}_0, \ldots, \mathcal{P}_s, \}$, then $\oplus\mathcal{P} = \mathcal{I}_s$, otherwise $\oplus\mathcal{P} = \mathcal{I}_\infty$. Figure 3 shows the construction of a minimal combination for the union of two trees.

Definition 4.13 [Path-Union] The *path-union* of two IMPs $\mathcal{I}, \mathcal{I}'$, denoted $\mathcal{I} + \mathcal{I}'$ is the unique IMP $\mathcal{I}^+$ such that $\mathsf{paths}(\mathcal{I}^+) = \oplus(\mathsf{paths}(\mathcal{I}) \cup \mathsf{paths}(\mathcal{I}'))$.

The *path-union* of two interactive presentation databases $\mathsf{ip}\mathcal{DB}, \mathsf{ip}\mathcal{DB}'$, denoted $\mathsf{ip}\mathcal{DB} + \mathsf{ip}\mathcal{DB}'$ is the path-union of the all IMPs in $\mathsf{ip}\mathcal{DB} \cup \mathsf{ip}\mathcal{DB}'$.

Definition 4.14 [Path-Intersection] The *path-intersection* of two IMPs $\mathcal{I}, \mathcal{I}'$, denoted $\mathcal{I} * \mathcal{I}'$ is the unique IMP $\mathcal{I}^*$ such that $\mathsf{paths}(\mathcal{I}^*) = \oplus(\mathsf{paths}(\mathcal{I}) \cap \mathsf{paths}(\mathcal{I}'))$.

The *path-intersection* of two interactive presentation databases $\mathsf{ip}\mathcal{DB}, \mathsf{ip}\mathcal{DB}'$, denoted $\mathsf{ip}\mathcal{DB} * \mathsf{ip}\mathcal{DB}'$ is the intersection of the all IMPs in $\mathsf{ip}\mathcal{DB} \cup \mathsf{ip}\mathcal{DB}'$.

The definition of path-difference, however, is a little bit more complex. Even though the difference of two IMPs is defined in the same way as shown above, the difference of two presentation databases is simply defined set theoretically.

Definition 4.15 [Path-Difference] The *path-difference* of two IMPs $\mathcal{I}, \mathcal{I}'$, denoted $\mathcal{I} \setminus \mathcal{I}'$ is the unique IMP $\mathcal{I}^\setminus$ such that $\mathsf{paths}(\mathcal{I}^\setminus) = \oplus(\mathsf{paths}(\mathcal{I}) \setminus \mathsf{paths}(\mathcal{I}'))$.

The *+-difference* and the **-difference* of two interactive presentation databases, $\mathsf{ip}\mathcal{DB}, \mathsf{ip}\mathcal{DB}'$ are given by:

$$\begin{aligned} \mathsf{ip}\mathcal{DB} \setminus_+ \mathsf{ip}\mathcal{DB}' &= +(\mathsf{ip}\mathcal{DB}) \setminus +(\mathsf{ip}\mathcal{DB}'). \\ \mathsf{ip}\mathcal{DB} \setminus_* \mathsf{ip}\mathcal{DB}' &= *(\mathsf{ip}\mathcal{DB}) \setminus *(\mathsf{ip}\mathcal{DB}'). \end{aligned}$$

Theorem 4.8 The operations of path-union, path-intersection and path-difference preserve both validity and strong validity.

The theorem is an immediate consequence that validity is a property of the content components of nodes, which are not effected by the operations of path-union, path-intersection and path-difference.

Theorem 4.9 Suppose $\mathsf{ip}\mathcal{DB}$ is an interactive presentation database containing $\mathcal{I}, \mathcal{I}'$. Then:

1. Selections may be pushed through path-unions, i.e. $\sigma_O[\mathsf{oc}](\mathcal{I} + \mathcal{I}') = \sigma_O[\mathsf{oc}](\mathcal{I}) + \sigma_O[\mathsf{oc}](\mathcal{I}')$.
2. Projections may be pushed through path-unions, i.e. $\pi_\wp[\mathsf{pc}](\mathcal{I} + \mathcal{I}') = \pi_\wp[\mathsf{pc}](\mathcal{I}) + \pi_\wp[\mathsf{pc}](\mathcal{I}')$.
3. Merge and join are both left and right distributive with respect to path-union.

The following result shows that similar equivalences hold for path-intersections.

Theorem 4.10 Suppose $\mathsf{ip}\mathcal{DB}$ is an interactive presentation database containing $\mathcal{I}, \mathcal{I}'$. Then:

1. Selections may not always be pushed through path-intersections, i.e. $\sigma_O[\mathsf{oc}](\mathcal{I} * \mathcal{I}') \subseteq \sigma_O[\mathsf{oc}](\mathcal{I}) * \sigma_O[\mathsf{oc}](\mathcal{I}')$.
2. Projections may be pushed through path-intersections, i.e. $\pi_\wp[\mathsf{pc}](\mathcal{I} * \mathcal{I}') = \pi_\wp[\mathsf{pc}](\mathcal{I}) * \pi_\wp[\mathsf{pc}](\mathcal{I}')$.
3. Merge is not distributive with respect to path-intersection:

- right distributivity:
  $\mathcal{I}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} (\mathcal{I}' * \mathcal{I}'')[\mathsf{wpc}_2] \subseteq$
  $(\mathcal{I}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}'[\mathsf{wpc}_2]) \cap (\mathcal{I}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}''[\mathsf{wpc}_2])$
- left distributivity:
  $(\mathcal{I} * \mathcal{I}')[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}''[\mathsf{wpc}_2] \supseteq$
  $(\mathcal{I}[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}''[\mathsf{wpc}_2]) \cap (\mathcal{I}'[\mathsf{wpc}_1] \lhd^{\mathsf{cm}}_{\wp_1:\wp_2} \mathcal{I}''[\mathsf{wpc}_2])$

4. Join is only left distributive with respect to path-intersection:
   - right distributivity:
     $\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} (\mathcal{I}' * \mathcal{I}'')[\mathsf{wpc}] \neq (\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} (\mathcal{I}')[\mathsf{wpc}]) * (\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} (\mathcal{I}'')[\mathsf{wpc}])$
   - left distributivity:
     $(\mathcal{I} * \mathcal{I}')[\mathsf{wnc}] \bowtie^a_{N:\wp} \mathcal{I}''[\mathsf{wpc}] = (\mathcal{I}[\mathsf{wnc}] \bowtie^a_{N:\wp} (\mathcal{I}'')[\mathsf{wpc}]) * (\mathcal{I}'[\mathsf{wnc}] \bowtie^a_{N:\wp} (\mathcal{I}'')[\mathsf{wpc}])$

Unfortunately, similar distribution results are not true for path-difference in their most general form, and counter examples exist.

Theorem 4.11 Suppose $\mathsf{ip}\mathcal{DB}$ is an interactive presentation database containing $\mathcal{I}, \mathcal{I}'$. Then:

1. Selections may not always be pushed through path-difference, i.e. $\sigma_O[\mathsf{oc}](\mathcal{I} \setminus \mathcal{I}') \subseteq \sigma_O[\mathsf{oc}](\mathcal{I}) \setminus \sigma_O[\mathsf{oc}](\mathcal{I}')$.

2. Projections may be pushed through path-difference, i.e. $\pi_\wp[\mathsf{pc}](\mathcal{I} \setminus \mathcal{I}') = \pi_\wp[\mathsf{pc}](\mathcal{I}) \setminus \pi_\wp[\mathsf{pc}](\mathcal{I}')$.

4.3 Set Theoretic Database Operations

In addition to the above operators, we include simple set based operation to the MPA algebra. These operators are in fact equivalents of the union, intersection and set difference operations in the relational algebra, but we use the prefix "set" to differentiate between path and set operations. Recall that in relational algebra, the set semantics is always preserved by removing the duplicates after every operation. In IPDs, duplicate elimination can be achieved at two levels, by removing duplicate paths or by removing duplicate trees. The set theoretic operations, set-union ($\cup$), set-intersect ($\cap$) and set-subtract ($-$) achieve the tree level duplicate elimination. These operators are defined in the obvious way.

Theorem 4.12 Selection and projection can be pushed down all set operations in MPA. In other words, for all unary operators $\langle op \rangle \in \{\sigma, \pi\}$ and for $\langle so \rangle \in \{\cup, \cap, -\}$, the following is true $\langle op \rangle (\mathsf{ip}\mathcal{DB} \langle so \rangle \mathsf{ip}\mathcal{DB}') = (\langle op \rangle\ \mathsf{ip}\mathcal{DB})\ \langle so \rangle\ (\langle op \rangle\ \mathsf{ip}\mathcal{DB}')$.

The theorem follows from the definition of the set oriented operators, and the definitions of the algebra operators applied to databases. Indeed, the unary operators applied to a set $\mathsf{ip}\mathcal{DB}$ return another set, obtained from $\mathsf{ip}\mathcal{DB}$ by applying the operator to every tree in $\mathsf{ip}\mathcal{DB}$, while the binary operators, given $\mathcal{I}$ and $\mathcal{I}'$, return a set containing the result of the combinations of all pairs in $\mathcal{I}$ Cartesian $\mathcal{I}'$. The result of a single algebraic operator on a tree is independent of the other trees possibly existing in the database.

5 Related Work

One of the first attempts to build multimedia presentations was due to Buchanan and Zellweger [4] who described a constraint based approach to specifying temporal aspects of a multimedia presentation. Their solution to the presentation constraints was based on the simplex algorithm. Kim and Song [8] extended this framework with elastic time, and provided improved algorithms. Later, Candan *et al.* [6] showed that instead of using arbitrary constraints, a small class of constraints called difference constraints was enough not only to specify temporal aspects of a multimedia presentation, but also to specify spatial aspects, together with a variety of quality of service (QoS) constraints. In [6], Candan et.al. further introduce *prioritized difference constraints* and they show how to relax presentation constraints (and modify the presentation in the most desirable manner) when there are conflicts within presentation specifications or there are resource limitations. Song [13] identified the appropriateness of difference constraints for specifying temporal aspects of a presentation at around the same time. Bertino and Ferrari [3] have developed an extension of the CHIMP based approach to perform detailed reasoning about individual media objects. Other related work includes [12, 10] which use intervals for scheduling multimedia presentations (interval constraints form a strict subclass of difference constraints). *None of these approaches deals with interaction.*

Work on the development of a *presentation database* has been pioneered by the Özsoyoglus' [9, 11] who have developed a graphical query calculus and a graphical query language. In their framework, a multimedia presentation is modeled as a graph where a node in the graph represents a media object. In contrast, our framework applies **(i)** even in the non-interactive case, **(ii)** allows querying of interactive multimedia presentations, **(iii)** provides an algebra that is specialized towards the querying of interactive multimedia presentations, while their framework is a calculus, **(iv)** our algebra contains unique operators like "merge" and "join", and **(v)** we present query equivalence results for query optimization. In addition, a large body of work [7] has concentrated on querying and constructing parts of Web which has an underlying tree structure. Most of this work uses higher level query languages which produce a single tree as output. Our language is not sophisticated enough to deal with

inter-document structures as well as intra-document links without the introduction of external methods. In contrast, these languages do not deal with temporal and spatial constraints which are specific to presentations. Finally, MPA makes a clear distinction between content and structure based operators in the algebra level making it easy to manipulate multiple presentations.

6 Conclusions

With the growing use of computers to create technical, business and artistic presentations, there has been a huge increase in the number of multimedia presentation tools out in the market. Almost all such presentations are interactive (e.g. PowerPoint requires hitting "return" to advance to the next slide and/or to display different bullets on a slide, while tools like ToolBook have different buttons connected to scripts which when hit, will invoke the script).

Thus, over the coming years, there will be a growing need to build the ability to query databases consisting of interactive multimedia presentations, to query this rapidly growing, but much neglected body of data [1]. In this paper, we have taken a first step (building upon previous important steps taken by the Özsoyoglus' [9, 11]) towards the definition of an interactive multimedia presentation database — in particular, we have:

- defined the concept of an interactive multimedia presentation database (IPD) that consists of static as well as dynamic interactive presentations;
- developed a set of algebraic operations on IPDs including very general versions of select, project and join type operations from the relational algebra;
- defined the concepts of validity, shown how validity may be preserved for different operations and developed methods for implementing different operations in the algebra;
- derived a host of query equivalence results that form rewrite rules to be used in a query optimization system for multimedia presentation databases.

In our ongoing work, we are **(i)** developing a multimedia presentation calculus, **(ii)** working on developing an implementation of the MPA on top of a relational database system and a PowerPoint presentation database, and **(iii)** developing cost models for the algebraic operators described here, so as to build a scalable query optimizer for multimedia presentation databases.

Acknowledgements

This work was supported by the Army Research Office under Grants DAAH-04-95-10174, DAAH-04-96-10297, and DAAH04-96-1-0398, by the Army Research Laboratory under contract number DAAL01-97-K0135, by an NSF Young Investigator award IRI-93-57756, and by a TASC/DARPA grant J09301S98061.

References

[1] S. Adalı (1998) *Making Peace with Your Multimedia.*, IEEE Intelligent Systems, 13(6), pp. 7–10.

[2] S. Adalı, M.L. Sapino, and V.S. Subrahmanian (1998) *Interactive Multimedia Presentation Databases, I: Algebra and Query Equivalences*, TR 98-04, Computer Science Dept., Renssealer Polytechnic Institute.

[3] E. Bertino, E. Ferrari. (1998) *Conceptual Temporal Models for Multimedia Data*, IEEE Trans. on Knowledge and Data Engineering, vol. 10, no.4, pp.612-631, July/August 1998.

[4] 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.

[5] 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.

[6] 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 Conf., Boston, MA, Nov. 1996.

[7] D. Florescu, A. Levy, and A. Mendelzon. (1998) *Database Techniques for the World-Wide Web: A Survey*, ACM SIGMOD Record 27:3, September 1998, pp. 59–74.

[8] M.Y. Kim and J. Song (1995) *Multimedia Documents with Elastic Time*, ACM Multimedia Conf. '95, 1995.

[9] J. Lin and Z.M. Özsoyoglu. (1997) *Processing OODB Queries by O-Algebra*, Proc. 8th Int. Conf. on Information and Knowledge Management, Rockville, Maryland, Nov. 1996, pp. 134-142.

[10] T.D.C. Little and A. Ghafoor (1990) *Synchronization and Storage Models for Multimedia Objects*, IEEE J. on Selected Areas of Communications, vol. 8, no. 3, April 1990, pp. 413-427. April 1990.

[11] G. Özsoyoglu, V. Hakkoymaz, and J.D. Kraft (1996) *Automating the Assembly of Presentations from Multimedia Databases*, Twelfth Int. Conf. on Data Engineering, pp. 593-601, New Orleans, February 1996.

[12] T.K. Shih, S.K.C. Lo, S.-J. Fu, and J.B. Chang (1996) *Using Interval Logic and Inference Rules for the Automatic Generation of Multimedia Presentations*, IEEE Int. Conf. on Multimedia Computing and Systems '96, pp. 425-428, Hiroshima, June 1996.

[13] J. Song, Y.N. Doganata, M.Y. Kim and A.N. Tantawi. (1997) *Modeling Timed User Interactions in Multimedia Documents*, in Proc. 1997 IEEE Intl. Conf. on Multimedia Computing Systems.

[14] V.S.Subrahmanian. (1998) "Principles of Multimedia Database Systems", Morgan Kaufmann.