A Specification Model for Temporal and Spatial Relations of Segments in Multimedia Presentations

Veli Hakkoymaz
Fatih University, Computer Engineering Department
Buyukcekmece / Istanbul, TURKEY
hakkoymaz @ fatih.edu.tr

Journal of Digital
Information Management

ABSTRACT: *In this paper, we introduce a conceptual tuple-based model for specification of spatial, temporal and semantic relationships among media segments in a multimedia presentation. A multimedia presentation contains various kinds of multimedia objects such as audio, video, images, text media types and so on. The organization of a presentation is a complex task in that the display order of presentation content (in time and space) must be specified. The critical decisions for presentation construction include content selection, content organization, and content delivery. Once the decision is made on the organization of the content of the presentation, it must be conveyed to the end user in the correct organizational order and in a timely fashion. Assuming that a set of multimedia segments gets organized into a presentation graph, this conceptual model formalizes a complete presentation using event-points for these segments. Three possible end-user environments of multimedia presentation are considered for presentation playout as (i) the one without any constraint, (ii) another one with a single constraint, and (iii) the third one with multiple constraints (one constraint for each type of multimedia segments in the organized presentation). In accordance with these limitations (i.e., without violating any of the end-user specified constraints), we give three methods for constructed presentations to be presented at the presentation terminal.*

Categories and Subject Descriptors
H.5.1 [Multimedia Information Systems]: 1.7 [Document and text processing]: **H.3.3 [Information Search and Retrieval]** Retrieval models

General Terms: Multimedia, Multimedia presentation, Tuple-based model , Multimedia segments

Keywords: Document Clustering, Termination Criterion, Bayesian Information Criterion (BIC), Dynamic Clustering, Document Retrieval, Performance

Received: 12 August 2009; Revised 19 September 2009; Accepted 28 September 2009

1. Introduction

Recent advances in high-speed networking technologies make it possible to create automated multimedia presentations from a pool of various types of multimedia resources and deliver them on demand to the presentation end-users. In recent years, since people with various background and diversity use the Internet as primary means of communication and collaboration, the volumes and types of electronically available data have increased dramatically. In order to generate, integrate, process, store, distribute and present these data, a wide spectrum of sophisticated multimedia applications are developed [1, 2, 4, 6, 11, 18, 21]. The purpose of this paper is to describe a conceptual model for the specification of content organization in time and space for multimedia presentations.

A multimedia presentation consists of a variety of individual media resources called segments. Multimedia segments have various presentation characteristics such as type, size, temporal duration, and so on. A scenario for a multimedia presentation specifies both spatial and temporal ordering of segments, and the way end-users interact with the application. Starting from the multimedia information capture until the presentation delivery, high performance tools are required for accessing, retrieving, manipulating, and storing these segments, for transferring and delivering them in a presentation terminal according to a playout order. A multimedia presentation consists of many types of multimedia segments such as texts, images, sounds and videos in various formats, whose organization and playout order are specified in terms of spatial and temporal constraints. These segments are retrieved (say, from a multimedia database) and presented on-demand in a presentation terminal [4, 13]. Multimedia presentations are widely used in such fields as healthcare, education, training, advertisement, entertainment, and so on [6, 21, 22]. In general, multimedia segments of a presentation are stored on a server and the actual presentation takes place at a presentation terminal.

The organization of presentations is a complex task in that the display order of presentation contents (in time and space) must be specified. The critical decisions for presentation construction include (1) what the contents are, and (2) how the contents are organized (i.e., some parts of audio and video may be temporally related and have to be presented in parallel; some other parts can only be presented after certain subjects are covered, etc), (3) once the decision is made on the organization of the contents of the presentation, it must be conveyed to the end user in the correct organizational order and in a timely fashion [6, 9]. In this paper, we introduce a conceptual model to specify time-based relationships among segments in a multimedia presentation for different end-user environments and compute (in an efficient manner) the features of the presentation graph.

After multimedia resources of interest are located, it has become more and more feasible to organize them into automated multimedia presentations [19, 21, 26]. A multimedia presentation usually involves a number of output devices such as speakers for audio, and monitor windows and large screens for text and video display. Unlike conventional applications, multimedia presentation has an additional dimension which associates the flow of real-time with the presentation of various data streams. When delivered separately, each media stream causes a distinctly perceivable effect that persists for certain duration of time at the output device. It is, however, the composite effects caused by the simultaneous display of all related streams that determine the successful progress of the presentation along the temporal axis. Therefore, a major requirement of any system supporting multimedia presentation is the need to provide means for composition of its contents and synchronization during multimedia presentation to maintain spatial and temporal relationships between the various data streams of multimedia objects. Furthermore, the need to provide the presentation end-users with the ability to exercise control (i.e., via his/her environment parameters) over multimedia presentations adds to the complexity of the multimedia presentation task. This research integrates the time-based specification into a multimedia system in order to express and visualize various presentation-related algorithms in a concise and unambiguous manner.

The rest of the paper is organized as follows: In Sec 2, after surveying the related work, we briefly describe a multimedia presentation system with its components. Sec 3 gives preliminary definitions and constructs used in the paper. In Sec 4, we introduce a model for representing a presentation graph of an organized presentation and then, give a temporal and spatial specification in a presentation. In Sec 5, three playout scenarios are given for constructed presentations for the environment of presentation end-users. Sec 6 discusses user control operations during the playout. Last section, Sec 7, provides the concluding remarks and direction of future work.

2. Related Work

Research works on preparing multimedia segments in a database into a multimedia presentation and conveyance of the resulting presentation to the presentation end-users have gained momentum in the literature [5, 6, 8, 9, 12, 21]. Multimedia segments in a specific subject are selected and organized as a presentation that is to be presented to the end-users. Based on J. F. Allen's work [17] that formalizes the relationships between temporal intervals, Little and Ghafoor [8] have developed a temporal-interval model (i.e., TIB) that captures the timing relationships among multimedia data segments. Inter-segment temporal relations are either imposed at the creation time of the multimedia segments (i.e., *live synchronization*) or set up artificially (i.e., *synthetic synchronization*). In their work, presentation of each multimedia data segment is represented by a time interval (start time, end time, duration). Using this model, they come up with a playout schedule for the segments with monotonically increasing deadlines in order to present them in a timely manner. To this extent, they specify the temporal access algorithms to facilitate forward and reverse playout as well as partial-interval evaluation for pause-resume operations. In work [12], three distinct problems in multimedia presentations are identified as determining the contents as well as the layout of the presentation in time and space. However, the main focus of the work is concerned with the description of temporal aspects of an abstract presentation behavior. Synchronization and control of temporally related actions are modeled by presentation frame types, sequentializer, parallelizer, splitter, combiner, and brancher.

Event-based features and layout facilities of a presentation are provided in *synchronized multimedia integration language* (SMIL), a declarative XML language for distribution of synchronized audio, video and other media in the presentation [7, 10]. Temporal and spatial specifications of media segments in a presentation are defined through the timing and layout constructs in SMIL. The SMIL standards, however, do not completely specify all aspects of the spatial and temporal composition of multimedia segments. For example, they don't address the issue of storage, retrieval, execution and sharing of applications. In addition, event modeling and composition schemes don't adequately cover the requirements for the variety of events that might occur in a multimedia application.

Presentation of multimedia objects implies their delivery to the presentation end-users, not their retrieval from a depository (database). In general, user interaction with a presentation must be supported through some well-defined operations so that users may influence the course of presentation actions [1, 3, 13, 21, 25]. Thus, the specification of temporal aspects of a presentation behavior takes into consideration the possible user inputs as well as the layout and synchronization issues of media data streams. The purpose of the organization is the temporal ordering of presentation actions and interactions. Having determined the actions in the presentation, the creator/user of the multimedia data has to specify the temporal relations between different actions, thereby defining a particular presentation order. Between two actions, there exists only one of 13 temporal relations: *before, meets, overlaps, during, starts, finishes*, inverse relations of these six relations and *equal* [8, 17].

In [15], temporal casual relationships (i.e., the end of video A causing the start of video B) between time intervals corresponding to multimedia objects are modeled. As for spatial specification, only two topological relationships (out of the 13 above), meet and overlap, are modeled. In addition, directional relationships of spatial specification are not handled between objects (i.e., A is right of B, A is 10 cm away from B). Formalism is introduced for executing/rendering multimedia application; however, the authoring is not formalized for specifying presentations. In [16], temporal specification of multimedia objects is either serial (one after another) or parallel (start or stop multiple objects at the same time). Objects are considered active while on the screen, and when they are removed from the screen, they are deactivated. As for the spatial relationships, presentation objects are considered as occupying a rectangular area, which are scaleable according to the needs of presentations. However, user interactions during a presentation execution such as pause and resume are not addressed. In [3], a methodology is introduced for temporal specification based on flexible playtime lengths of presentation objects, which are specified as a pair of values. Presentation duration of multimedia objects can be stretched or shrunk between these two values for meeting their time constraints during their presentations.

Presentation authoring is a non-trivial task since there are a large number of multimedia segments to consider for selection and afterwards among these selected segments one has to specify organization constraints for both temporal and spatial orders [1, 2, 4, 14]. In [1, 5], an authoring methodology is defined as well as a set of integrity checking tools for developing multimedia applications. The primary objective is a spatiotemporal composition model for the multimedia application design. Checking for integrity is performed on multimedia application scenarios. Spatiotemporal composition of a presentation is specified through declarative means in terms of spatial and temporal ordering of multimedia objects. In other words, a multimedia presentation is specified in terms of

- temporal layout of segments (i.e., temporal duration and relationship of each segment in the presentation)
- spatial layout of segments in application window (i.e., for each time point in the presentation)
- playout of multimedia segments for presentation end-users.

A multimedia presentation system has the functionality of content selection, organization and playout of multimedia presentations. The content selection component selects a set of multimedia resources on the local database as well as multimedia segments from the Web, which are expanded into a larger conceptually coherent set of linked information resources, based on similarities and close relationships among concepts of these resources (i.e., these concepts are present in their corresponding metadata files). In the presentation organization component, we use a fixed set of organization constraints to specify organization relations among these selected set of multimedia segments. In the presentation playout component, by satisfying a set of user specified parameters, multimedia segments from the organized presentation are played out in correct organizational order for the presentation end-users.

3. Presentation content organization

Presentation content can be selected by some automated tools using adlets / intelligent agents [18, 20, 26] or a constraint-driven methodology based on the inclusion / exclusion constraints [19, 23]. Content organization is specified by a presentation author using the organization constraints [6]. When the content is organized, a presentation graph will result representing the organized presentation.

3.1 Presentation graph features

A *presentation graph* G=(V,E) is a directed acyclic graph (i.e., a dag) augmented with two special nodes, *initial node* I (i.e., *the source*) and the *final node* F (i.e., *the sink*), where nodes in V(G) are labeled with the segments in the presentation, and edges in E(G) indicate the relative presentation order of two segments (i.e., $a \rightarrow b$ specifies that segment a is ``before'' segment b in the presentation). Except for the initial and final nodes, each node in the presentation graph has at least one incoming edge and one outgoing edge. Edges are added to a presentation graph according to the specified organization constraints. Associated with each node of a presentation graph is a start time (a computed attribute), a finish time (a computed attribute), a time duration or length (an essential attribute), a multimedia segment (object to be presented), and clip id together with its duration (can replace a full segment with its clip in order to meet a deadline). A presentation graph has a *length* (i.e., the length of the longest path from its source to its sink, that is, the sum of the lengths of all segments on the longest path) as well as a *height* (i.e., the maximum number of concurrently playing segments, that is, the maximum cut in a temporally-aligned graph [Liu68]). The formal ways to determine the height and length of the presentation graph are described in [4, 6]. Let SD=(s_1,s_2,..., s_n) denote the selected set of multimedia segments that are to appear in the presentation. Let OC=(o_1,o_2,...,o_t) denote the presentation organization constraints that are specified among multimedia segments in SD. Given any two sets SD and OC, the subpresentations are constructed in two stages [6]. In the first stage, the organizationally related segments are grouped together according to the organization constraints. In the second stage, we augment each group with two extra nodes, *initial node* I and *final node* F and a number of edges. A directed edge from initial node I to every node that do not have an incoming edge is added. Similarly, we add a directed edge from every node in the group that do not have an outgoing edge to the final node F. In this way, each group becomes a subpresentation. Nodes I and F represent the start and terminate nodes, respectively, for each subpresentation. They both are empty (null) segments. After the augmentation, we call each graph G_i a *subpresentation*. Using the described procedure, we form a unique presentation graph from a given set of segments and organization constraints (for details, see [6]).

Then, after obtaining an event-point representation of this presentation graph (i.e., event-point representation is explained in Sec. IV.A), we carry out the spatial specification for the presentation by way of *spatial operations*. These stages are listed as:

Stage 1: Determine the content of the presentation.

Stage 2: Complete the temporal specification through organization constraints.

Stage 3: Obtain the event point representation of the presentation graph.

Stage 4: Complete the spatial specification through spatial constraints.

4. Event point Model for presentation graphs

In this section, for a given presentation graph, we show how to generate its event-point representation. In this way, we specify composition of a presentation through declarative means. In short, we specify (i) temporal layout of segments (i.e., temporal duration and relationship of each segment in the presentation), (ii) spatial layout of segments in application window (i.e., for each time point in the presentation), and (iii) playout order of multimedia segments for application end-users.

4.1. Event-Based Representation of Presentation Graph

Let us call the start and the end of each media segment in a multimedia presentation as the *start* and the *end* event or simply *"event"*. Assuming that there are n segments in a presentation, the total number of events is 2n. Events occur at specific points in time, called *event points*. The total number of *event points* can be fewer than the total number of events since multiple events may occur at the same event point. Therefore, the total number of event points is at most 2n. Event points can be sorted in ascending or descending order since they are time values (i.e., positive numbers). For a multimedia presentation, let m denote the total number of event points, where $m \le 2n$, and let *collection* C_i denote the group of events that occur at the i^{th} event point, $1 \le i \le m$.

Formally, a *collection C* will be characterized by a 4-tuple [E, S, T, A] where E is the event point (i.e., a positive number), S is the set of multimedia segments that start at E, T is the set of multimedia segments that end at E, and A is the set of active multimedia segments (i.e., currently playing out at the terminal). At first sight, the information on active segments seems redundant; however, they include both those segments which have already started but not ended yet, and those to start to play out at event point E). After ordering all the collections in increasing order of event points (i.e., with respect to E), we associate each collection with an index like $[E, S, T, A]_i$ for the i^{th} collection, $1 \le i \le m$, or an element of a collection C_i as E_i, S_i, T_i, and A_i.

The event-based representation can be useful in specifying several properties (i.e. constraints) with respect to its presentation. We express these as follows:

Property 1 (Equality): Since all the segments that start will eventually end, the following equality holds:

$$\sum_{i=1}^{m} |\, Si \,| = \sum_{i=1}^{m} |\, Ti \,|$$

Property 2 (Active Segments): Set of active segments at any event point E_k in the presentation is given by the expression below:

$$A_k = \bigcup_{i=1}^{k} Si - \bigcup_{i=1}^{k} Ti$$

Property 3 (Active Segments via Recursion): By setting A_0=0, the set of active segments at event point E_k can be computed in an efficient manner (i.e., recursively) by the formula below ($1 \le k \le m$):

$$A_k = (S_k \bigcup A_{k-1}) - T_k$$

Property 4 (Presentation-Height-at-Interval): Presentation height between two consecutive event points E_k and E_{k+1}, k<m can be computed using the following formula:

$$h_k = \sum_{i=1}^{k} | Si | - \sum_{i=1}^{k} | Ti |$$

This equals to the size of the set of active segments at event point E_k. In other words, $h_k = |A_k|$. In other words, the presentation height refers to the number of concurrently playing segments of a presentation. At any point in time (i.e., say at time *d*) it equals the presentation height between event point of collection j and event point of collection j+1 where $j \le d < j+1$ and $j < m$. This is true because no new edge appears or disappears in the presentation graph between two consecutive event points.

Property 5 (Presentation Height): For a presentation graph G,

the *presentation height* $hG = \max_{i=1 \text{ to } n} \{|Ai|\}$

4.2 Temporal Specification

We specify a multimedia presentation by defining composition of participating media segments in time and space along with a real-time axis. In other words, this specification describes both temporal and spatial ordering of segments. Instead of synchronization, we use the term *composition* to represent both temporal and spatial ordering of segments. In composition architecture, action refers to the presentation of a segment that participates in multimedia applications. Temporal specifications are given relative to some reference point, which refers to presentation of other multimedia segments in the preceding, current, or succeeding event collections. To illustrate, the following two examples are provided.

Example 1: Figure 1 depicts a sample presentation graph in a timeline diagram. Using the notations introduced, we can characterize this presentation with the following event collections (n=8;m=6):

Collection C_1 : [0, {a}, {}, {a}]

Collection C_2 : [1, {b,c}, {a}, {b, c}]

Collection C_3 : [2, {d, e}, {b}, {c, d, e}]

Collection C_4 : [3, {g,f}, {c,d}, {e, g, f}]

Collection C_5 : [4, {h}, {g,e}, {f, h}]

Collection C_6 : [5, {}, {h,f}, {}]

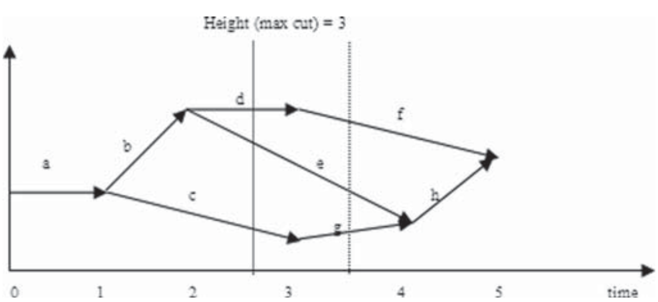


Figure 1. A timeline diagram showing the event points of a presentation graph

Using the property 4 *(i.e., Presentation-Height-at-Interval),* given above, the heights between two consecutive event points for this presentation graph are computed as follows:

height at [0,1) = 1 - 0 = 1;
height at [3,4) = 7 - 4 = 3;

height at [1,2) = 3 - 1 = 2;
height at [4,5) = 8 - 6 = 2;

height at [2,3) = 5 - 2 = 3;
height at [5, ∞)= 8 - 8 = 0;

Thus, by *Property 5 (i.e., Presentation Heigh),* the height of this presentation graph is 3.

Example 2: A verbal description of a multimedia presentation, called *News*, taken from [1], is given as: "The News starts with presentation of image A (located at point 50,50 relative to the application origin Θ). At the same time, background music E starts. Ten seconds later video clip B starts. It appears to the right side (18 cm) and below the upper side of A (12 cm). Just after the end of B, another multimedia application related to fashion (Fashion_clip) starts. Fashion_clip consists of a video clip C that shows the highlights of a fashion show and appears 7 cm below (and left aligned to) the position of B. A narration N appears right below and left aligned to the clip C during its presentation. Three seconds after the start of C, a text logo D (the designer's logo) appears inside C, 8 cm above the bottom side of C, aligned to the right side. D will remain for 4 seconds on the screen. Meanwhile, at the tenth second of the News clip, the TV channel logo (F) appears at the bottom-left corner of the application window. F disappears after 3 seconds. The application ends when music background E ends." Temporal presentation ordering of segments in the News is depicted in the timeline diagram in Figure 2. The presentation *News* consists of eight event collections in the following temporal order: C_1; C_2; C_3; C_4; C_5; C_6; C_7; C_8. Each collection can be specified in event-based representation as given below:

Collection C_1 = [0, {A, E}, {}, {A, E}]

Collection C_2 = [10, {B, F}, {}, {A, B, E, F}]

Collection C_3 = [13, {}, {F}, {A, B, E}]

Collection C_4 = [17, {C, N}, {B}, {A, C, E, N}]

Collection C_5 = [20, {D}, {}, {A, C, D, E, N}]

Collection C_6 = [24, {}, {D}, {A, C, E, N}]

Collection C_7 = [28, {}, {C, N}, {A, E}]

Collection C_8 = [32, {}, {A, E}, {}].

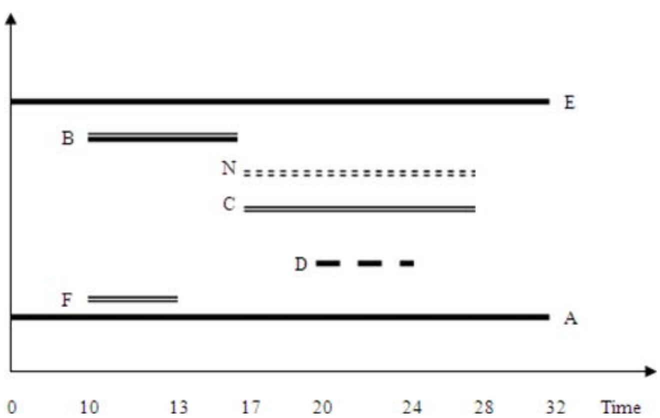


Figure 2. Temporal ordering of segments in News

```
Algorithm compute_event_points(G, s)
        Input: Presentation graph G with its initial node S
        Output: Playout start and finish times of each node in G
          for each node v in G
                start_time[v] ← 0
          insert s into node_queue
          start_time[s] ← 0; finish_time[s] ← duration[s]
          while node_queue is not empty do
             remove a node u from node_queue
             for each node v to which there is a directed edge from u
                 if (start_time[v] < start_time[u] – duration[u]) then
                             start_time[v] ← start_time[u] – duration[u]
                     finish_time[v] ← start_time[v] – duration[v]
                              insert v at the end of node_queue
                      return G with start and finish times for each node
```

Figure 3. Pseudo code for computing event points of a presentation graph

As illustrated in the two examples above, for a presentation of m event points, the temporal characterization of a multimedia presentation can be specified by a sequence of event collections in this order, C_1, C_2, …, C_m. This event point characterization can be very useful for a multimedia presentation system. First of all, the proposed model allows one to express the playout behavior of segments in a presentation in a concise and unambiguous way. In addition, it can be utilized to express various presentation-related algorithms. It helps visualize the presentation behavior of segments. Also, we can derive various features of the presentation using the event point representation of the presentation graph. Figure 3 shows such an example for computing the event points in a given presentation:

4.3 Spatial Specification

We specify spatial composition of segments in a multimedia presentation through spatial layout and relationships of participating segments with respect to each other in the application. Three spatial aspects are identified between two multimedia segments. These are listed as follows:

1) topological relationships between objects: disjoint, meet, overlap and so on
2) directional relationships between objects: left, right, above, above-left and so on
3) distance characteristics between objects: outside 5cm, inside 2cm, and so on

We express first two of these aspects with spatial constraints (i.e., to be defined shortly) and the third one with two distance arguments (i.e., H for horizontal and V for vertical distances). We define spatial constraints to express both topological and directional relationships between two segments for each event point in the presentation and their distance characteristics are indicated with distance arguments to the spatial constraints. By default, these arguments are assumed to have zero values.

The framework for spatiotemporal composition model consists of spatial specifications of segments for each event point in the presentation. Spatial specifications of multimedia segments (say, in collection C_i) are given relative to a reference point, which can be listed as (i) Θ: spatial start of the application (i.e., upper left corner of application window), (ii) segments in the active set, or (iii) multimedia objects in previous collections, C_1, C_2, …, C_{i-1}.

For a presentation, spatial specification is performed only after both the temporal specification is carried out through organization constraints and event point representation of presentation graph gets determined. Once these two preconditions are satisfied, we use spatial constraints (i.e., to be defined shortly) to complete the spatial specification of the multimedia presentation. Figure 4 lists the procedural steps for defining spatial relations of segments in a presentation where Ai is set of active multimedia segments at event point Ei in event-based model.

```
A0 ← {}
for each event point i ←1 to m do /* consider collection Ci in order */
  newSet ← Ai - Ai-1 /*newSet is set of newly emerging segments at event point i in
presentation */
  if newSet is not empty then // item:a rectangular, top, right, bottom, left sides -
clockwise
   for j = 1 to |newSet| do
     specify spatial constraint for segment j in newSet with respect to a reference point:
            (a) some segments in Ai,
            (b) Θ: spatial start of application,
            (c) some segments in previous collections, C1, C2, …, Ci-1
```

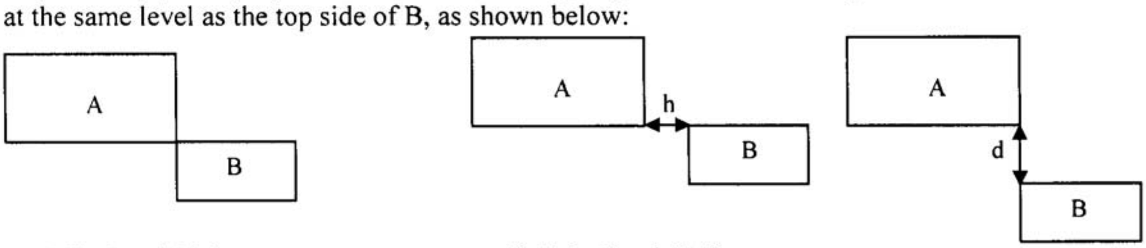
Figure 4. Pseudo code for specifying spatial relations of segments based on event points for a presentation graph

4.4 Spatial Constraints

Spatial relations between any two segments are defined by constraints of this subsection. Each subpresentation represented with an organized presentation graph can be viewed as a rectangular shape with a height and a length. They occupy certain portions of display screen of presentation terminal, which is determined by the spatial constraints. In general, two multimedia resources appearing in the same event point can have the following spatial relations with each other. Let A and B denote two different segments/subpesentations, each representing a rectangular shape.

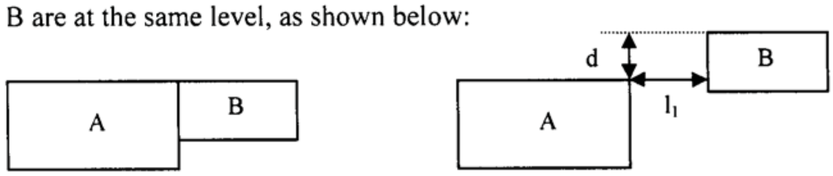
i) A and B can be apart from each other (in several different ways, H for horizontally or V for vertically):

- *HApartAtTop(A, "left-of", B, H, V)*: Horizontally, segment A is left of segment B and the bottom side of A is at the same level as the top side of B, as shown below:



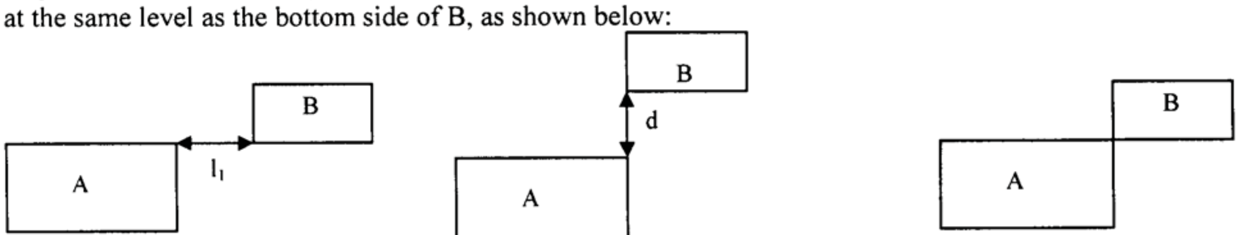
a) H= 0 and V=0 b) H=h >0 and V=0 c) H= 0 and V=d >0

- *HApartAtLevel(A, "left-of", B, H, V):* Horizontally, segment A is left of segment B and the top sides of A and B are at the same level, as shown below:



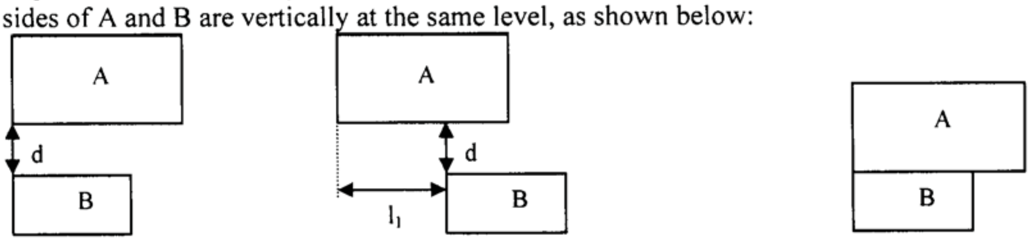
a) H= 0 and V=0 b) $H=l_1 > 0$ and $V=d > 0$

- *HApartAtBottom(A, "left-of", B, H, V)***:** Horizontally, segment A is left of segment B and the top side of A is at the same level as the bottom side of B, as shown below:



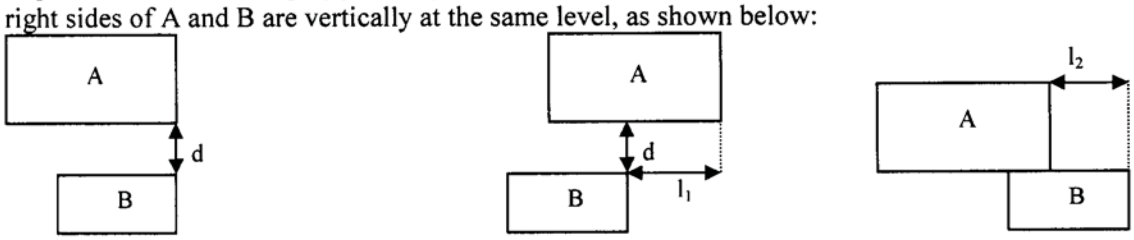
a) $H=l_1 > 0$ and V=0 b) H=0 and $V=d > 0$ c) H=0 and V=0

- *VApartAtLevel(A, "on-top-of-from-left", B, H, V):* Vertically, segment A is on top of segment B and the left sides of A and B are vertically at the same level, as shown below:



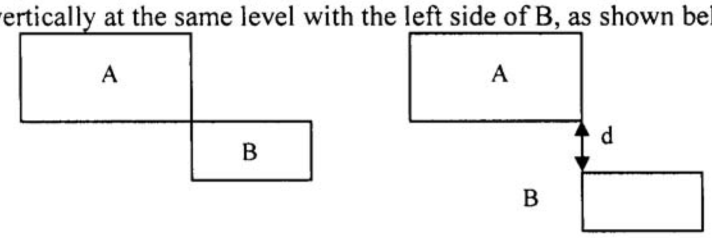
a) H=0 and $V=d>0$ b) $H=l_1>0$ and $V=d>0$ c) H=0 and V=0

- *VApartAtLevel(A, "on-top-of-from-right", B, H, V):* Vertically, segment A is on top of segment B and the right sides of A and B are vertically at the same level, as shown below:



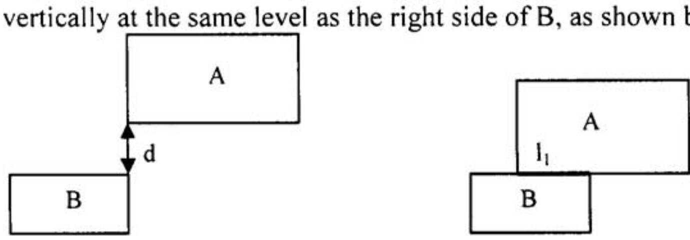
a) H=0 and $V=d > 0$ b) $H=l_1 < 0$ and $V=d > 0$ c) $H=l_2 > 0$ and V=0

- *VApartAtLeft(A, "on-top-of", B, H, V):* Vertically, segment A is on top of segment B and the right side of A is vertically at the same level with the left side of B, as shown below:



a) H=0 and V=0 b) H=0 and $V=d >0$

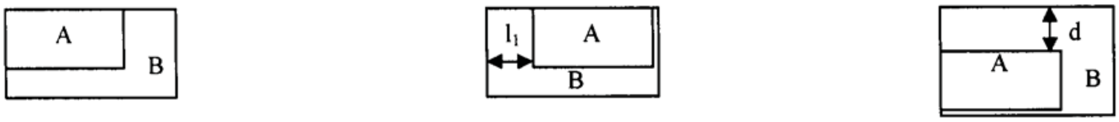
- *VApartAtRight(A, "on-top-of", B, H, V):* Vertically, segment A is on top of segment B and left side of A is vertically at the same level as the right side of B, as shown below:



a) H=0 and $V=d > 0$ b) $H=l_1 > 0$ and V=0

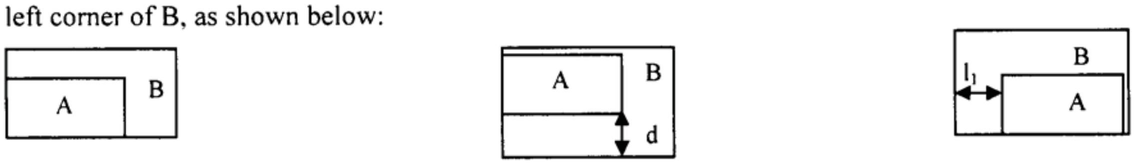
ii) A and B can be overlapping. (One is inside of the other, in several different ways).

- *InsideAtTop(A, "left-of", B, H, V):* Segment A is inside of segment B in that A is overlapping on top-left corner of B, as shown below:



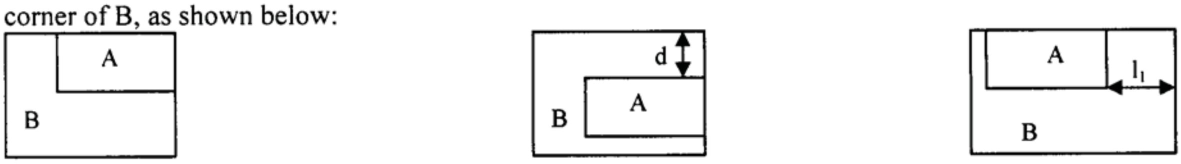
a) H=0 and V=0 b) H= l_1>0 and V=0 c) H=0 and V=d>0

- *InsideAtBottom(A, "left-of", B, H, V):* Segment A is inside of segment B in that A is overlapping on bottom-left corner of B, as shown below:



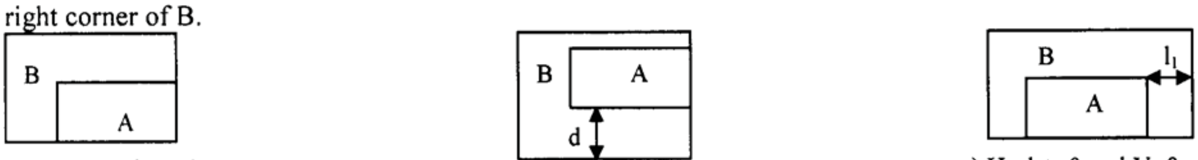
a) H=0 and V=0 b) H=0 and V=d>0 c) H=l_1>0 and V=0

- *InsideAtTop(A, "right-of", B, H, V):* Segment A is inside of segment B in that A is overlapping on top-right corner of B, as shown below:



a) H=0 and V=0 b) H=0 and V=d > 0 c) H= l_1 > 0 and V=0

- *InsideAtBottom(A, "right-of", B, H, V):* Segment A is inside of segment B in that A is overlapping on bottom-right corner of B.



a) H=0 and V=0 c) H= l_1 > 0 and V=0

b) H=0 and V=d>0

Having defined this set of spatial constraints and their semantics, we now provide an example:

Example 3: Spatial specification of the News presentation is as follows:

The News clip starts with presentation of image A (located at 50,50 relative to the application origin Θ).

InsideAtTop(A, "left-of", Θ, H=50 pts, V=50 pts)
At the same time, background music E starts.
InsideAtTop(E, "left-of", Θ, H=0 pts, V=0 pts)
Ten seconds later video clip B starts. It appears to the right side (18 cm) and below the upper side of A (12 cm).
HApartAtLevel(A, "left-of", B, H=18 cm, V= - 12 cm)
Just after the end of B, another multimedia application related to fashion (Fashion_clip) starts. Fashion_clip consists of a video clip C that shows the highlights of a fashion show and appears 7 cm below (and left aligned to) the position of B.
VApartAtLevel(B, "on-top-of-from-left", C in Fashion_clip, H=0, V=7 cm)

A narration N appears right below and left aligned to the clip C during its presentation.

VApartAtLevel(C in Fashion_clip, "on-top-of-from-left", N, H=0, V=0)

Three seconds after the start of C, a text logo D (the designer's logo) appears inside C, 8 cm above the bottom side of C, aligned to the right side. D will remain for 4 seconds on the screen.

InsideAtBottom(D, "right-of", C in Fashion_clip, H=0, V=8 cm)

Meanwhile, at the tenth second of the News clip, the TV channel logo (F) appears at the bottom-left corner of the application window. F disappears after 3 seconds. The application ends when music background E ends."
InsideAtBottom(F, "left-of", Θ, H=0, V=0)

5. Presentation Environment

Presentation author can construct multiple multimedia presentations for the same subject but with different presentation properties. For example, a presentation with certain height and length property can be calculated and organized. Using a compact representation of multimedia segments (i.e., media clips [13]), another version of the same presentation with a shorter presentation length can be prepared. Another presentation with the same subject can have certain height limit. Still another one with the same subject can have certain height limit for each type of multimedia segments found in the presentation. In short, in an environment of presentation assembly and organization, given a set of various types of multimedia segments, among which organization constraints are defined by some subject experts, one can come up with various multimedia presentations (consisting of these segments) with different properties. Each of these presentations is represented by a different presentation graph structure.

To illustrate, let Pr_1 represent a multimedia presentation about a certain topic (say, *education*) with a calculated height and length. Similarly, Pr_2 may represent another multimedia presentation about the same topic with a shorter length and height. Yet another presentation, Pr_3, on the same *education* subject excludes certain types of multimedia segments from its contents. This concept of exclusion can be generalized in yet another presentation (say, Pr_4) with the same subject, which has a specific height for each type of the multimedia segments that it contains. When a presentation end-user request arrives with playout parameters specifying properties of the end-user's environment, most appropriate presentation among these gets selected for the requesting end-user's site and played out at a presentation terminal.

5.1 Playing Out Constructed Presentation

This section presents three models for how presentations are played out according to the end-user's presentation environment. In an end-user environment, there are a maximum number of available presentation mediums (i.e., resources) for each of media types. For example, there can be two resources for presenting audio sources, 4 windows for presenting video streams, 3 screens for presenting text and so on. To illustrate, for a user, his/her environment may have no constraints on the number of presentation resources, another user environment may have just some general constraints represented in the user profile. Constraints in yet another user environment may be specified manually by its end-user. Among many choices of various multimedia presentations satisfying an end-user's subject requirements, the most appropriate one gets selected with respect to a matching scheme between playout constraints of the end-user's environment and presentation properties maintained for each presentation. The following models illustrate how the presentation graph represented as a collection of events are utilized in order to visualize the playout of involved segments in the presentation terminal for end-users. Recall that an event collection is a 4-tuple [E, S, T, A] where E is the event point, S is the set of multimedia segments that start at E, T is the set of multimedia segments that end at E, and A is set of active multimedia segments at event point E. After ordering all the collections in increasing order of event points (i.e., with respect to E), we associate each collection with an index like $[E, S, T, A]_i$ for the i^{th} collection, $1 \leq i \leq m$, or an element of a collection C_i as E_i, S_i, T_i, and A_i.

*User Environment Model1 (Presentation Playout Without any Constraint)***:** In this model, no height limitation is specified by the end-user of multimedia presentations. In other words, the presentation terminal at the end-user environment has as many presentation resources as the presentation graph requires. To illustrate, the pseudo-code in Figure 5 shows presentation behavior of multimedia segments in this model.

User Environment Model2 (*Presentation with a Single Constraint*)**:** In this model, a single height limitation is specified by

```
Height←0; alreadyStarted←0; alreadyEnded←0;
for i←1 to m do
    Start presentation of segments in Si
alreadyStarted←alreadyStarted + |Si|
Terminate presentation of segments in Ti
    alreadyEnded←alreadyEnded + |Ti|
    CurrentHeight← alreadyStarted -
alreadyEnded
    if CurrentHeight > Height then
       Height← CurrentHeight
end
```

Figure 5. Pseudo code for presentation of multimedia segments if an end-user has specified no constraint

the end-user of the presentation. At the maximum, the end-user specified UH*eight* many concurrent media segments are played out at the presentation terminal. In other words, presentation terminal has at most UH*eight* many presentation resources for playing out the multimedia presentation represented by the presentation graph. The pseudo-code in Figure 6 is provided as a sample for this model.

```
Height←0; alreadyStarted←0; alreadyEnded←0;
for i←1 to m do
  if[(|Si|+alreadyStarted)-
  (|Ti|+alreadyEnded)] ≤ UHeight then
     start presentation of segments in Si
     alreadyStarted←alreadyStarted + |Si|
  Terminate presentation of segments in Ti
     alreadyEnded←alreadyEnded + |Ti|
     CurrentHeight ← alreadyStarted -
     alreadyEnded
     if CurrentHeight > Height then
        Height ← CurrentHeight
else /* exceeds limit if all starts in Si */
      Terminate presentation of segments in Ti
     alreadyEnded←alreadyEnded + |Ti|
     delta ← UHeight -
     [(|Si|+alreadyStarted)-alreadyEnded]
     Start presentations of as many as delta
     segments in Si
     alreadyStarted ← alreadyStarted + delta
     CurrentHeight ← alreadyStarted -
     alreadyEnded
     if CurrentHeight > Height then
       Height ← CurrentHeight
end /* of for */
```

Figure 6. Pseudo code for presentation of multimedia segments if end-user has specified a single height constraint

User Environment Model3 (*Presentation with Multiple Constraints*): In this model, the end-user of the presentation specifies the height limitation for each of the media types appearing in the presentation graph. In accordance with these limitations (i.e., without violating any of the end-user specified limits), the presentation graph is presented at the presentation terminal. Figure 7 shows the pseudo-code for this model.

```
Height[1..max_type]←0
alreadyStarted[1..max_type]←0;
alreadyEnded[1..max_type]←0;
for i←1 to m do
  for j←1 to max_type do
    TS←subset of Si whose type is of type j
    TE←subset of Ti whose type is of type j
    if[(|TS|+alreadyStarted[j])-
    (|TE|+alreadyEnded[j])]≤ UHeight[j] then
      start presentation of segments in TS
      alreadyStarted[j]←alreadyStarted[j] +
      |TS|
   Terminate presentation of segments in TE
      alreadyEnded[j]←alreadyEnded[j] + |TE|
      heightNow[j]←alreadyStarted[j]-
      alreadyEnded[j]
      if heightNow[j] > Height[j] then
        update new height as
        Height[j]←heightNow[j]
    else /* type j exceeds limit if all
    starts in TS */
    Terminate presentation of segments in TE
      alreadyEnded[j]←alreadyEnded[j]+|TE|
      delta←UHeight[j]-
```

```
      (|SE|+alreadyStarted[j])-alreadyEnded
      Start presentations of delta many
      segments in SE
      alreadyStarted[j]←alreadyStarted[j]+delta
      heightNow[j]← alreadyStarted[j]-
alreadyEnded[j]
      if heightNow[j]> Height[j] then
        Height[j]←heightNow[j]
   /* end of inner for */
/* end of outer for */
```

Figure 7. Pseudo code for presentation behavior of multimedia segments if end-user has specified multiple constraints

5.2 Evaluation of Playout Algorithms

Three playout algorithms based on event point representation of the presentation graph are empirically evaluated and compared. As shown in these algorithms, the proposed model allows one to express the playout behavior of segments in a presentation in a concise and unambiguous way. In addition, it can be utilized to express various presentation-related algorithms. It helps visualize the presentation behavior of segments. We can derive various features of the presentation using the event point representation of the presentation graph. In the first model, end-user has not specified a height limitation for his/her presentation environment. In other words, presentation terminal has as many presentation resources as the presentation graph required. In the second model, a single height limitation is specified by the end-user of the presentation. At the maximum, collectively, the end-user specified *height* many concurrent media segments are played out the presentation terminal. In other words, presentation terminal has at most *height* many presentation resources for playing out the presentation represented by the presentation graph. In this case, the end-user makes no distinction among types of the segments to be presented.

In the third model, the end-user of the presentation specifies the height limitation for each of the multimedia types appearing in the presentation graph. In accordance with these limitations (i.e., without violating any of the end-user specified limits), the presentation graph is presented at the presentation terminal. As we said before these limitations may be specified by the end-user or can be obtained automatically from the end-user's presentation environment.We are implementing a prototype for each of end-user environment models for presentation system.

6. User Control Durıng Playout

To allow interactive user control over the playout of an organized presentation, an arbitrary number of control buttons are designated in the multimedia presentation system. Through these buttons, a user is able to change/affect the flow of the presentation in certain ways. This section introduces such a control mechanism. Interactivity lets the multimedia presentation system react in response to the request of a user in presentation playout time and adjust and reorganize (if necessary) the ongoing presentation accordingly. In addition to the usual control operations, such as *start presentation playout* and *stop presentation playout*, we provide a number of additional user control operations that may introduce delays to an already playing presentation and, thus, requiring dynamic reorganization of the presentation graph so as to fit the presentation to the given playout length and height constraints.

The control operations provided to the user are categorized into two groups: Global control operations and local control operations. Global control operations affect the overall presentation while local control operations are applied to the individual windows where single streams are played out but may cause organizational changes on the overall presentation. User activates the control operations, for example, by clicking on the push-buttons dedicated for each control operation on the screen using a pointing device such as a mouse. Local control buttons are displayed under screen windows where individual streams are played out. On the other hand, global conrol buttons are displayed on a seperate area on the screen where the overall presentation is played out.

6.1 Local Control Operations

- **Suspend**: Pressing on the ``Suspend'' button under an individual window where a multimedia stream is being played out and entering the value y in the dialog box that is displayed right after this activation has the effect that the regarding stream is frozen for y time units.
- **Pause and Play**: Pressing on the ``Pause'' button under an individual window has the effect that the stream that is being played out in the window is paused until the ``Play'' button of the regarding window is pressed on.

6.2 Global Control Operations

- **Suspend:** Pressing on the "Suspend'' button of the overall presentation and entering the value y in the displayed dialog box has the effect that the overall presentation, i.e. all streams being played out in individual windows, is frozen for y time units.
- **Pause and Play :** Pressing on the ``Pause'' button of the overall presentation has the effect that the overall presentation is frozen until the ``Play'' button of the overall presentation is pressed on. During this intervall all streams are frozen in all windows.
- **Rewind** : Pressing on the "Rewind" button of the overall presentation and entering the value y in the displayed dialog box has the effect that the overall presentation, i.e. all streams being played, is rewound to a collection C_i where event points satisfy $E_{i-1} < t \leq E_i$ and presentation starts with segments in S_i , followed by segments from subsequent collections.
- **Rewind to a Concept**: Pressing on the "Rewind to a Concept" button of the overall presentation and selecting a concept (i.e., **person gets in** a **vehicle** at t=8 in Figure 8) among a collection of concepts in a choice box has the effect that the overall presentation, i.e. all streams being played, is rewound to a collection C_i where event points satisfy $E_{i-1} < t \leq E_i$ and presentation starts with segments in S_i, followed by segments from subsequent collections.

7. Conclusion and Future Work

In this paper, we have described a specification model for a multimedia presentation in terms of time-, space-related aspects and user-specific controls. We have shown how the event-point representation of the presentation graph helps us to elegantly express the playout behavior of multimedia segments for end-user environments as (a) without any constraint, (b) with a single constraint, and (c) with multiple constraints (one constraint for each type of multimedia segments in the organized presentation). For each of these environments, in accordance with limitations (without violating any of end-user specified constraints), the presentation is carried out at the presentation terminal.

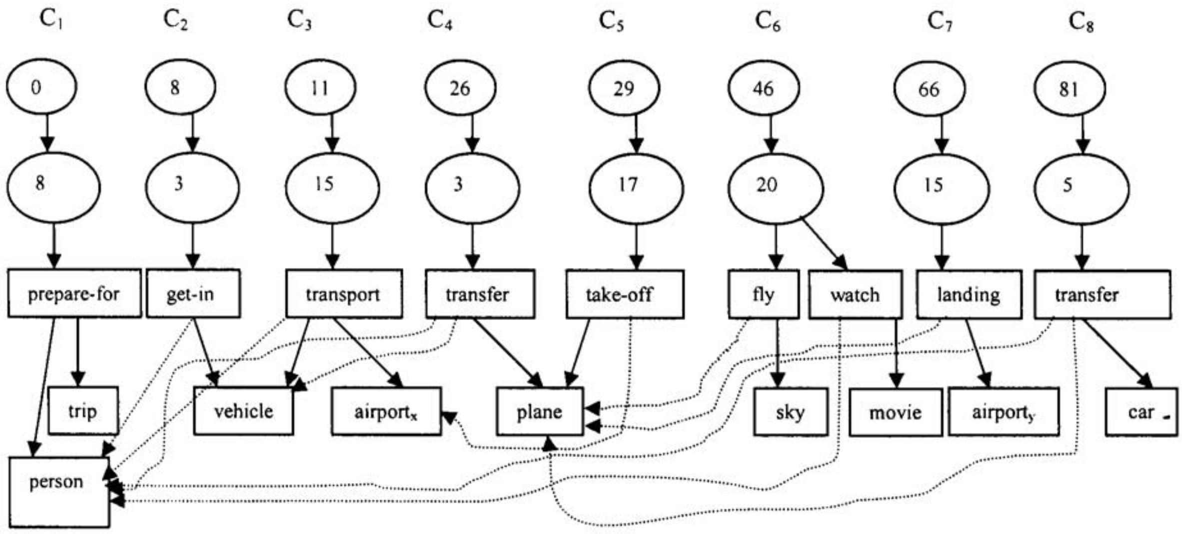


Figure 8. Representation of concepts for collections of event points in a multimedia presentation. Collections are listed in the top row. Below the collections are the corresponding event points, each of which points to the duration of "a concept", i.e. person prepares for a trip in 8 time units

References

[1] Vazirgiannis, M., Kostalas, I., Sellis. T. (1999). Specifying and Authoring Multimedia Scenarios, *IEEE Multimedia*, 6 (3) 24-37. July-September

[2] Jourdan, M., Layaida, N., Roisin, C. (1998). A Survey on Authoring Techniques for Temporal Scenarios of Multimedia Documents, *Handbook of Internet and Multimedia Systems and Applications, Part 1:Tools and Standards,* CRC Press, Apr.

[3]. Kim, M.Y., Song, J. (1995). Multimedia Documents with Elastic Time, *Proc. of ACM Multimedia 95,* 143-154. Nov.

[4] Hakkoymaz, Veli (1997). A Constraint-Driven Methodology for Designing a Multimedia Presentation System from Multimedia Databases, PhD thesis, Case Western Reserve Univ., Cleveland, OH.

[5] Vazirgiannis, M., Theodoridis, Y., Sellis, T. (1998). Spatio-Temporal Composition and Indexing for Large Multimedia Applications, *ACM/Springer Multimedia Systems,* 6 (4) 284-298, July.

[6] Veli Hakkoymaz., Ozsoyoglu, G. (1997). A Constraint-Driven Approach to Automate the Organization and Playout of Presentations in Multimedia Databases, *J. of Multimedia Tools and Applications*, 4, 171-197

[7] SYMM Working Group, (2001). Synchronized Multimedia Integration Language (SMIL 2.0), Available: http://www.w3.org/TR/SMIL20.

[8] Little, T. D. C., Arif Ghafoor. (1993). Interval Based Conceptual Model for Time Dependent Multimedia Data, *IEEE Trans. on Knowledge and Data Eng*, 5 (4) 551-563.

[9] Little, T. D. C. (1993). A Framework for Synchronous Delivery of Time-Dependent Multimedia Data, *Multimedia Systems*, 1, 87-94.

[10] Newman, R. M., Gaura, E. I. (2003). Formal Design of SMIL Presentations, ACM SIGDOC'03, San Francisco,CA.

[11] Bulterman, D. C. A. (2003). Using SMIL to Encode Interactive Peer-Level Multimedia Annotations, ACM DocEng'03, Grenoble, France.

[12] Hoepner, P. (1991). Synchronizing the Presentation of Multimedia Objects, ACM SIGOIS 12 (1) 19-31.

[13] Veli Hakkoymaz, Znati., F., Dogdu, E. (1997). User Control and Dynamic Reorganization of Multimedia Presentations during Playout, *Int. Journal of Software Eng. and Knowledge Eng.*, V.7 (3).

[14] Alpaslan Altunkopru, (2002). Language Features for Multimedia Presentation Specification on Multimedia Database, Thesis, M. S, Dept. of Computer Eng. Fatih University, Istanbul, Turkey.

[15] Iino, M., Day, Y. F., Ghafoor, A. (1994). An Object-Oriented Model for Spatiotemporal Synchronization of Multimedia Information, Proc. of the IEEE Multimedia Conf, IEEE CS Press, 10-119.

[16] Yu, J., Xiang, Y. (1997). Hypermedia Presentation and Authoring System, Proc. of 6th Int'l WWW, April, 153-164.

[17] Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals, *Comm. of the ACM*, 26, Nov, 832-843.

[18] Chang, S-K., Znati, T. (2001). Adlet: An Active Document Abstraction for Multimedia Information Fusion, *IEEE Trans. on Knowledge and Data Engineering*, 13 (1) Jan/Feb, 112-122.

[19] Veli Hakkoymaz, Kraft, J., Ozsoyoglu, G. (1999). Constraint-Based Automation of Multimedia Presentation Assembly, *Multimedia Systems,* v.7 (6) Dec, 500-518.

[20] Hendler, J. (2001). Agents and Semantic Web, *IEEE Intelligent Systems*, 16 (2) 30-37.

[21] Santanche, A., Medeiros, C. B., Pastorello, G. Z. (2007). User-Author Centered Multimedia Building Blocks, *Multimedia Systems*, 12, 403-421.

[22] Combi, C., Oliboni, B., Rossato, R. (2005). Merging multimedia presentations and semistructured temporal data: a graph-based model and its application to clinical information, *Artificial Intelligence in Medicine*, v.34 (2).

[23] Ozsoyoglu, G., Hakkoymaz, V., Kraft, J. (1996). Automating the assembly of presentations from multimedia databases, IEEE Int'l Conf on Data Eng, Feb, 593-601.

[24] Dick, C. A., Bulterman., Lynda Hardman. (2005). Structured multimedia authoring, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), v.1 (1) 89-109, February.

[25] Chen, S.-C., Sista, S., Shyu, M.-L., Kashyap, R.L. (2000). An Indexing and Searching Structure for Multimedia Database Systems, Proc. of IS&T/SPIE Conf. on Storage and Retrieval for Media Databases, 262-270.

[26] Veli Hakkoymaz. (2009). Multimedia Presentation Organization and Playout Management using Intelligent Agents, Springer *Journal of Multimedia Tools and Applications*.