Multimedia Documents with Elastic Time

Michelle Y. Kim *Junehwa Song* *

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

YOON, SONG @ WATSON.IBM.COM

Abstract

Time is an essential component in interactive multimedia documents (or systems). We present the *elastic time model* for multimedia documents. Using the metaphor of a spring system, it allows authors to associate with each multimedia object a minimum and a maximum length and a length at rest. Authors can connect the (elastic) objects by defining temporal relationships among them. If the given specification is consistent, a document is produced which is also elastic, with a minimum, a maximum, and an optimal length. As such, our elastic model associates with a document a range of feasible solutions in addition to an optimal one. The author can then select from the acceptable range an alternative length for the document, and the system will compute a revised solution that takes the additional global constraint into effect. The system can answer questions, such as: "Can I show this multimedia presentation in 10 minutes? If so, how should all the objects be scheduled?" Furthermore, the system also takes fairness into consideration and distributes any necessary stretching or shrinking across multimedia objects contained in a document. As such, the elastic time model provides expressive power and flexibility in document authoring and browsing. The proposed approach has been implemented in Smalltalk/OS2.

KEYWORDS: Multimedia documents, multimedia authoring, elastic time model, temporal constraint systems, linear programming.

1 Introduction

Multimedia authoring is a process of ordering a set of multimedia objects, such as video segments, images, graphics, audio segments, or text segments. Ordering of objects can be done in the temporal dimension as well as in the spatial dimension. In this paper, we discuss the problem of obtaining a temporal layout, or a schedule.

Time is an essential dimension in a multimedia system. Temporal synchronization issues in authoring multimedia documents have motivated considerable research activities [6, 17, 9, 14, 2, 3, 5]. Reasoning about time has also been the focus of much of research activities within Artificial Intelligence, and many formalisms have been proposed for temporal reasoning [1, 4]. Most notably, James Allen's interval algebra [1] has received much attention for its simplicity. Allen introduced an interval-based temporal logic based on 13 possible relationships between a pair of intervals, and proposed a reasoning algorithm based on constraint propagation.

Allen-style constraint networks, which deal only with qualitative information, have been integrated with metric information in a constraint-based reasoning system in [4]. The system allows uncertainties in the representa-

*The author's current address is: University of Maryland, College Park, MD 20742, JUNESONG @ CS.UMD.EDU

tion of time. Events are represented by time intervals, each of which is bounded by its minimum duration and the maximum duration. If the network (of events) is consistent, the system produces two sets of answers: a set of earliest possible times and a set of latest possible times. The algorithm is polynomial, but it is limited in that the nature of the solutions is extreme. The solutions consist of times that tend to be either minimum or maximum.

In recent years, the notion of stretching and/or shrinking durations has been explored to obtain a temporal layout of a multimedia document [3, 8, 19]. In the Firefly system [3], each media item contains two or more events: the start, end and possibly some internal events. The duration between two temporally adjacent events within a media item is represented by a triple of three values: minimum, optimum, and maximum durations. Media items provide separate stretching and shrinking costs that specify the penalty for selecting a duration other than the optimum. An author can connect a pair of events by a temporal relationship. Given the temporal constraints, the scheduler solves the problem of assigning times to the events using the linear programming technique. A system such as this can deal with various media items with some flexibility. But it is limited in that it computes only one solution for a given set of constraints. To find another consistent solution that might exist, authors must alter individual stretching and shrinking costs and solve the problem again.

In this paper, we describe the elastic time model that deals with temporal uncertainties while avoiding the extreme bounding values. It provides multimedia documents with flexibility at various levels of integration. It produces an optimal solution while stretching or shrinking the objects and satisfying the constraints (as in [3]). In addition, it provides authors with a range of solutions, giving them the capabilities for interactively refining the design of the document. If desired, authors can select from the range of solutions an alternative length for the document, and the system will compute a revised solution. The system can answer questions such as: "Given these objects and the relationships, can I show this multimedia presentation in 10 minutes? If so, what is the best way?" Furthermore, the elastic time model also provides a framework for achieving fairness in distributing necessary stretching or shrinking across the multimedia objects. This paradigm of authoring with elastic time has been implemented as part of the Isis authoring environment [10, 12, 16]. Details of the technical background may be found elsewhere [10, 16], and in this paper we focus on the elastic time model.

The next section describes elastic stories. Four temporal relations are defined. A simple example is given. In Section 3, we formulate part of our problem as a minimization problem in linear programming to obtain a minimal cost solution. In Section 4, we address the "fairness" issue in scheduling multimedia objects. A short discussion is provided summarizing our experiences in Section 5, and conclusions are given in Section 6.

2 Elastic Stories

Consider a multimedia document consisting of a number of multimedia objects, which we will call a *story*. Suppose that it takes 20 minutes to present, but it may take 10 minutes if everything is shown at its fastest speed, and 25 minutes at the slowest speed, and suppose that we have only 15 minutes to deliver the material. Since 15 falls into the acceptable range, 10 - 25, we know that there is a solution, which will be a bit faster than the usual 20 minutes. The challenge is to obtain a solution by stretching or shrinking the speeds of the objects within their defined ranges, while keeping them as close as possible to their optimal speeds. Note that stretching or shrinking does not always mean altering the speed; it may also be achieved by adjusting the length, e.g., some portions of audio can be skipped, or the last frame of a video

can be frozen for some time.

We view a multimedia document as a set of connected springs, where each spring is stretched or shrunk according to the forces given. As a spring has tendency to return to its length at rest, a multimedia object may stretch or shrink when necessary and by a smallest degree possible. Then, building a story involves defining (elastic) time intervals and connecting them by using temporal relations. In the rest of this section, we describe the temporal specification of elastic stories and their scheduling for presentation. We use a simple example to illustrate the advantages of using the elastic time model.

2.1 Temporal Specification

We base our specification language on Allen's temporal algebra[1], and take time intervals as primitives. To capture metric information and to provide flexibility in dealing with time, we associate with each time interval a triple of lengths. Associating with a time interval a triple of time estimates has a long history. PERT (Program Evaluation and Review Technique[13]), a classical project management tool, starts with three time estimates: most optimistic, most pessimistic, and most likely. They are then combined algebraically to a single value, to the expected elapsed time, using a weighted average. PERT enables users to see where they can save time and where they can let the schedule slide.

Let a multimedia object m be associated with a triple of lengths, or durations:

$$(\alpha_m, \lambda_m, \omega_m),$$
$$\alpha_m \leq \lambda_m \leq \omega_m,$$

such that m may be presented over a time interval I_m whose length falls in the range bounded by a minimum α_m and a maximum ω_m, and that λ_m is the most desirable, or optimal length. We call this triple three spring constants for obvious reasons. We say that object m is *stretchable* if $\alpha_m < \omega_m$, and *fixed* otherwise.

We now give a basic set of primitive relations that can hold between any number of objects:

- *co-start*$(m1, \ldots, mn)$: time intervals $I_{m1}, \ldots, I_{mn}$ share the same beginnings,
- *co-end*$(m1, \ldots, mn)$: time intervals $I_{m1}, \ldots, I_{mn}$ share the same ends,
- *meet*$(m1, \ldots, mn)$: time intervals I_{mi} is before I_{mj}, $i, j = 1, \ldots, n, i < j$, and if $i + 1 = j$, the end of I_{mi} is shared by I_{mj} as its beginning.

Note that m_i can be a "dummy" object such that it has nothing to show or play, but it may be associated with a range of non-zero intervals. Such an object can be used to cause a time delay. With this, relationships such as *overlap* can also be expressed using the above primitives.

In addition to the above three relations we add the following relation:

- *co-occur*$(m1, m2)$: time interval I_{mi} is set to be the same as that of I_{mj}.

Certain types of multimedia objects, such as images or text, have no intrinsic time intervals associated with them. Let m_i be such an object. Then m_i can be included as part of a story by setting the corresponding range of intervals. This can be done either by explicitly specifying a minimum, a maximum, and an optimal value for the object, or by using the *co-occur* relation. The *co-occur*(m_i, m_j) relation means both *co-start*(m_i, m_j) and *co-end*(m_i, m_j). More, certain objects may have associated with them flexible time lengths; the *co-occur* relation can also be used to synchronize those objects either by stretching or compressing them.

See Figure 1 for a graphical illustration of the four temporal relations. We represent multimedia objects visually using the simple building block metaphor. Multimedia objects are treated as electronic building blocks, or time-boxes, such that the lengths of the time-boxes are proportional to the (optimal) lengths of the corresponding objects. Two short video objects, "sneeze" and "cough", are represented by

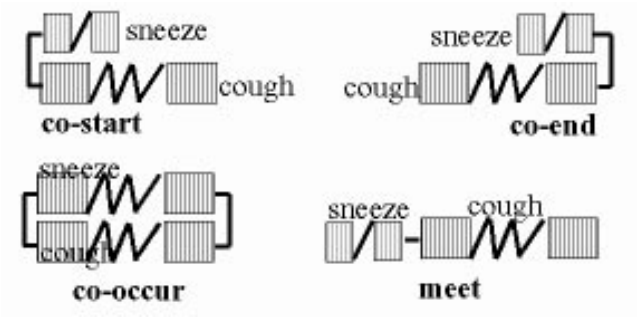

Figure 1: Four temporal relations for building elastic stories

two elastic time-boxes, with the following spring constants:

	min	opt	max
sneeze	3.4	3.4	7.0
cough	5.0	6.8	9.0

They can start together (*co-start*: indicated by connecting their left ends with a bracket); end together (*co-end*: indicated by connecting their right ends); start together and end together (*co-occur*: indicated by connecting both ends), or one can follow the other (*meet*). As "sneeze" is shorter than "cough" and is stretchable, the *co-occur* resulted in stretching the "sneeze" to the same length as the "cough".

A time interval can be viewed as a pair of two end-points. We call the two end-points of I_m, $start(I_m)$ and $end(I_m)$, where $start(I_m) < end(I_m)$. The values of $start(I_m)$ and $end(I_m)$ are relative with respect to the corresponding story S. We say that a time interval I_m is *instantiated* with respect to a story S, when $start(I_m)$ and $end(I_m)$ are set with respect to S.

Then, given a set of objects M for a story S, to find an "optimal-showing" of S is to find for all $m \in M$ corresponding instantiations which minimally deviate from their optimal lengths while satisfying all the constraints.

2.2 An Example

Consider a multimedia story with the following set of objects and the corresponding spring constants (in seconds):

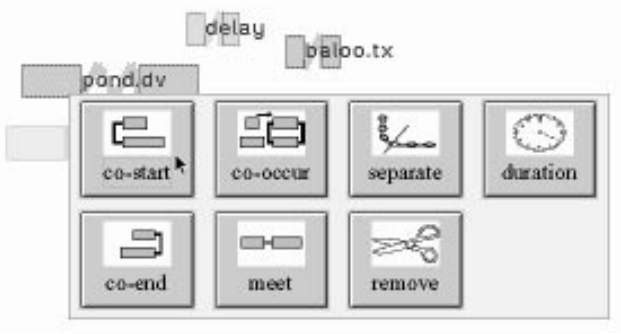

Figure 2: Selecting the *co-start* relation on "pond"

	Type	Min	Opt	Max
baloo	text	3.0	3.0	15.0
pond	video	7.0	9.9	14.0
delay	time delay	1.0	3.0	5.0
waltz	sound	10.0	10.0	10.0
bear	drawing	3.0	3.0	20.0

The following relationships are given among them:

co-start	(bear, delay)
meet	(delay, baloo)
co-end	(pond, baloo)
co-start	(pond, waltz)
co-end	(waltz, bear)

The graphical user interface for the Isis system allows authors to directly manipulate time-boxes; authors can drag and drop time-boxes, resize, connect, separate, or remove them. The system also provides a menu-based interface for simplicity. To relate "pond" and "waltz" by the relation *co-start*, the author simply touches the "pond" time-box to get a menu of choices: the four relations, *separate*, *remove*, and *duration* (Figure 2). The choice *duration* allows the author to modify the spring constants. Upon touching the *co-start*, the menu disappears, and a time-box with a question mark(?) appears (Figure 3). As the author touches the "waltz" box, it replaces the "?" box, and the *co-start*

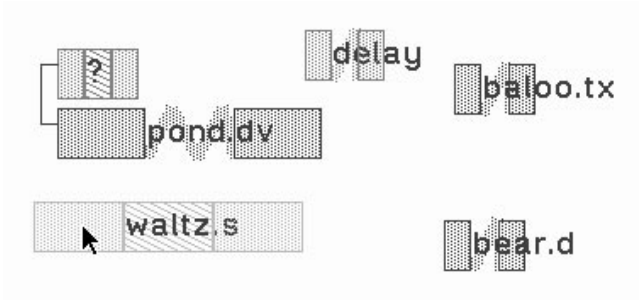

Figure 3: Relating "pond" and "waltz" by *co-start*

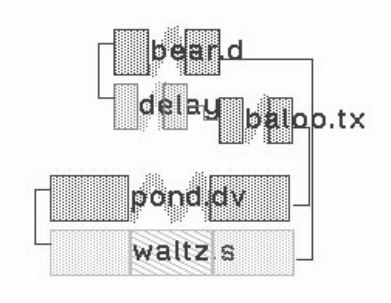

Figure 4: The optimal (10 second) solution of the story

relationship is established between "pond" and "waltz".

Because there is a solution that satisfies all the constraints, a story is created. The time-box representation of the optimal solution of the story "Baloo the bear" is shown in Figure 4.[1] As summarized in Table 1, it takes 10 seconds to present the optimal instance of the story. The lower and the upper bounds on the durations of the story are also obtained, 10 and 20 seconds, respectively. In the example, the lower bound happened to coincide with the optimum. If desired, the author can then choose a new solution within the range and refine the design of the document, as we shall see shortly.

2.3 Solutions for an Elastic Story

A solution to a constraint satisfaction problem is an assignment of values to all variables satisfying all constraints. The problem is to find one or all solutions. Our goal is to find a range of solutions bounded by a minimum and a maximum, and also identify an optimum solution. We let an optimum solution be an assignment of times to objects, such that the sum of the

[1]Note that the time-box layout algorithm we use provides a visual approximation of the corresponding objects. Unfortunately, our current algorithm cannot always align the time-boxes accurately, although all depicted constraints have been satisfied.

Figure 5: Author can reset the length of a story

amounts stretched or shrunk is minimal. Thus, the system can answer questions such as:

> What is the shortest, longest, and optimal time it takes to present this story?
>
> Can this story be presented in time T? If so, what is the best way?

Returning to our example, our method finds the triple of lengths (10, 10, 20) for the story "Baloo the bear"; it will take a minimum of 10 seconds, a maximum of 20 seconds to play the story, and ideally, it will take 10 seconds. But the author can shrink the story or stretch it (Figure 5). Setting the length of the story to 15 seconds (Figure 6), a new schedule (or instance) of the story is now available. The time-box representation of the new instance of the story is shown in Figure 7. Compare the two instances of the story in Figure 4 and Figure 7,

	10 second		15 second	
	length	offset	length	offset
baloo	3.0	6.9	7.1	4.9
pond	9.9	0.0	7.0	5.0
delay	1.0	5.9	4.9	0.0
waltz	10.0	0.0	10.0	5.0
bear	4.1	5.9	15.0	0.0

Table 1: 10 and 15 second solutions as shown in Figure 4 and in Figure 7

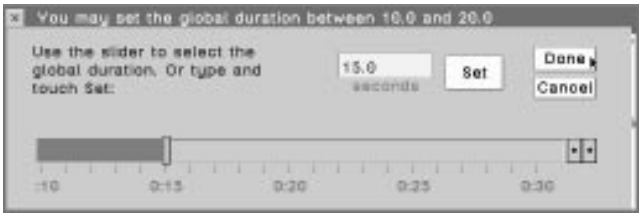

Figure 6: Adjusting the duration of the story

one without an added global constraint and the other with a global constraint added, respectively. According to our optimality criteria, the 10 second solution is the optimal solution, and the 15 second solution costs more, in the sense that the objects deviate more from their optimum lengths. But the 15 second solution might provide a more suitable design under certain situations.

Let M be a set of multimedia objects, and R be a set of temporal constraints given among M. Given M and R, the problem of producing a story S is to obtain the spring constants $(\alpha_S, \lambda_S, \omega_S)$ for S, i.e., the optimal duration and the upper and the lower bounds on the durations. Furthermore, the problem is also to obtain for a time T, such that $\alpha_S \leq T \leq \omega_S$, a schedule of story S, or an instance of S of length T, such that the schedule is consistent with the given constraints.

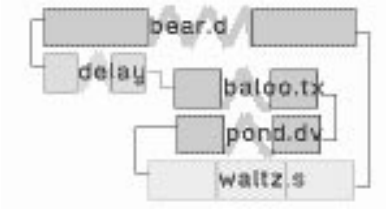

Figure 7: The 15 second solution of the story with a global constraint

A story is modeled by the temporal constraint network reflecting the temporal relationships and the minimum and the maximum durations of the objects as in [4, 10]. The system uses the all-pairs shortest paths algorithm, which computes between any two time points the minimum and the maximum possible duration. The lower bound, α_S, can be obtained by taking the maximum of all such minimum durations, and the upper bound, ω_S, by taking the maximum of all such maximum durations.

An optimum duration, λ_S, is the duration of the minimum cost schedule. A method for obtaining a minimal cost (optimal) solution is described in Section 3. Section 3.4 describes a method of obtaining a solution with a global constraint T added.

3 Minimal Cost Scheduling

In this section we present an approach to solve the problem of obtaining the optimal solution for an elastic story. We formulate the problem as a minimization problem in linear programming. We generate a set of algebraic equations from a given set of temporal relationships among the objects. The equations are then used as constraints in solving the minimization problem. A method is described that allows authors to reset the length of a story that has already been generated.

3.1 Formulating the Problem

Let S be a multimedia story defined by a set of multimedia objects M, and a set of tem-

poral relations R among M. Furthermore, let $(\alpha_m, \lambda_m, \omega_m)$ denote the spring constants associated with multimedia object $m \in M$. For each object m, a cost function is associated. Let $cost(I_m)$ denote the cost to play an object m for I_m time unit:

$$cost(I_m) = \begin{cases} |I_m - \lambda_m| & \text{if } \alpha_m \leq I_m \leq \omega_m \\ \infty, & \text{otherwise.} \end{cases}$$

Then our problem is to minimize the total sum of costs:

$$cost(I_S) = \sum_{m=1}^{n} cost(I_m). \qquad (1)$$

We have associated a cost function with each I_m for m, and that the solution we are after has associated with it $cost(I_S)$ which minimizes the sum of all $cost(I_m)$. Note that we have made a simplifying assumption that the cost is uniform across media types. This assumption can change and the cost function can be made subject to certain nonlinear costs, or relative weight as in [3].

3.2 The Temporal Graph

We proceed by representing the objects and the temporal constraints in R as a directed graph, which we call a *temporal graph*. From the temporal graph, we obtain a set of algebraic equations that we use as constraints in solving the minimization problem.

The temporal graph for story S, denoted by $G(S)$, consisting of a set of nodes $N = M \cup M'$ and a set of directed edges E, can be constructed as shown below. The sets Δ and Σ in the algorithm are used to identify the first and the last events of the story for the purpose of constraining the global length in Section 3.4.

1. For each relationship in R, do the following:
   - for each $meet(m_i, m_j) \in R$, add an edge $(m_i \rightarrow m_j)$ to E,
   - for each *co-start*(m_i, m_j), add two edges $(m_k \rightarrow m_i)$, $(m_k \rightarrow m_j)$ to E if there exists m_k such that $meet(m_k, m_i)$ or $meet(m_k, m_j)$. Otherwise, m_k be a null node with zero lengths $(0, 0, 0)$, add it to M', and add the two edges $(m_k \rightarrow m_i)$, $(m_k \rightarrow m_j)$ to E.
   - for each *co-end*(m_i, m_j), add two edges $(m_i \rightarrow m_k)$, $(m_j \rightarrow m_k)$ to E if there is m_k such that $meet(m_i, m_k)$ or $meet(m_j, m_k)$. Otherwise, let m_k be a null node with zero lengths $(0, 0, 0)$, add it to M', and add the two edges $(m_k \rightarrow m_i)$, $(m_k \rightarrow m_j)$ to E.
   - for each *co-occur*(m_i, m_j), treat it the same way as *co-start*(m_i, m_j) and *co-end*(m_i, m_j).
2. Add *head* and *tail* and relevant nodes:
   - construct Δ a set of starting objects: for each node m_k which has indegree 0, insert it to Δ if m_k is a null object (with zero lengths). Otherwise insert a null object, $start.null_k$, to Δ and to M', and insert an edge,$(start.null_k \rightarrow m_k)$ to E.
   - construct Σ a set of ending objects: for each node, m_k, which has outdegree 0, insert it to Σ if m_k is a null object. Otherwise insert a null object, $end.null_k$, to Σ and to M' , and insert an edge, $(m_k \rightarrow end.null_k)$ to E.
   - for each object in Δ and Σ, associate a triple of lengths, $(0, 0, \infty)$.
   - add a node *head* to M', and $(head \rightarrow m_k)$, for each $m_k \in \Delta$ to E. Similarly, add a node *tail* to M', and $(m_k \rightarrow tail)$ for each $m_k \in \Sigma$ to E. Both are of length (0, 0, 0).

In the temporal graph, the direction of the arrow represents the flow of time. The graph thus constructed does not have directed cycles. If there is one, it means that the configuration is not consistent since time is not reversible. The temporal graph of the story "Baloo the bear" in

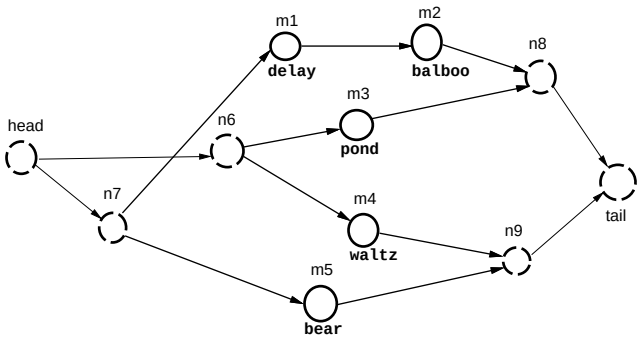

Figure 8: Temporal Graph

Section 2.2 is shown in Figure 8. Each solid circle, $m_1, ..., m_5$, represents the media objects in the story, i.e. delay,..., bear. Each dashed circle represents the null objects inserted to construct the graph. Here the set Δ is $\{n_6, n_7\}$ and the set Σ is $\{n_8, n_9\}$.

3.3 Solving the Problem

To find a minimum cost scheduling for a given story S, we generate a set of equations from the corresponding temporal graph $G(S)$. We generate an equation

$$X_i + \cdots + X_k = X_l + \cdots + X_n \qquad (2)$$

if $m_i \cdots m_k$ and $m_l \cdots m_n$ are paths in the graph $G(S)$, such that m_i and m_l share a parent, and m_k and m_n share a child. From the graph in Figure 8, the following equations can be generated.

- $X_1 + X_2 + X_8 = X_5 + X_9$
  (paths m_1, m_2, n_8 and m_5, n_9 share n_7 as parent and *tail* as child)
- $X_3 + X_8 = X_4 + X_9$
  (m_3 and m_4 share n_6 as parent, and n_8 and n_9 share *tail* as child)
- $X_6 + X_4 = X_7 + X_5$
  (n_6 and n_7 share the parent *head*, and m_4 and m_5 share the child n_9)
- $X_7 + X_1 + X_2 = X_6 + X_3$
  (n_7 and n_6 share the parent *head* and m_2 and m_3 share the child n_8)

Then our problem is to find a vector $X(X_1 \cdots X_n)$ which minimizes

$$cost(I_S) = |X_1 - \lambda_1| + \cdots + |X_n - \lambda_n| \qquad (3)$$

subject to all the equations generated according to Equation 2 and

$$\alpha_i \leq X_i \leq \omega_i, 1 \leq i \leq n.$$

The nonlinear objective function (3) with absolute value terms can be transformed to a linear form as follows [7].

For each X_i in the total cost function (3),

$$X_i - \lambda_i = A_i - B_i,\ A_i,\ B_i \geq 0.$$

We can easily prove that

$$min\{\sum |X_i - \lambda_i|\} = min\{\sum (A_i + B_i)\}.$$

We then solve the minimization problem of the new objective function, $\{\sum (A_i + B_i)\}$, using the simplex method [15] and obtain the minimum total cost $cost(I_S)$, along with a vector $X(X_1 \cdots X_n)$, where X_i is the length of object m_i.

3.4 With Global Constraints

Now we return to our question:

> "Given M and R, is there a story S which we can present in time T?"

To answer this question, we need to constrain the total length (global length) of the story. To constrain this global length, we need to identify in the story the first and last events of the story (i.e., the objects that start at the earliest time and the object that end at the latest time). However, it is not straightforward since the objects in the story are elastic. Consider the example in Section 2.2. Either "bear", "pond", or "waltz" can start at the earliest time, although

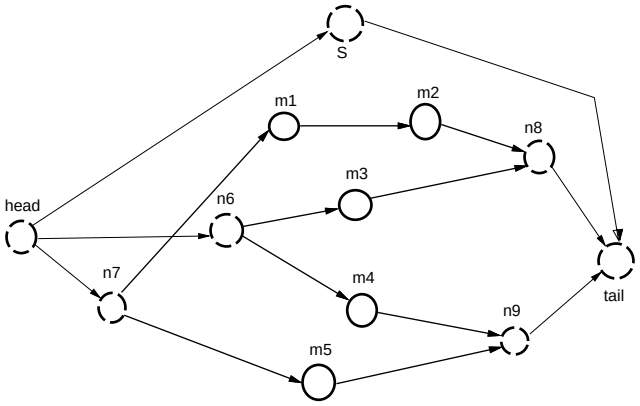

Figure 9: Temporal graph with global constraint

we don't know which one because they are elastic. Figure 4 shows an instance where "pond" and "waltz" are the two earliest ones and Figure 7 shows an instance where "bear" is the earliest.

The author can set the global time T, and reset the spring constants $(\alpha_S, \lambda_S, \omega_S)$ of story S to:

$$(x, T, y), \text{ where } \alpha_S \leq x \leq T \leq y \leq \omega_S.$$

A node representing the new instance of story S is added (Figure 9) to the temporal graph by adding: $(head \rightarrow S)$ and $(S \rightarrow tail)$.

We now proceed to solve the problem of obtaining a minimum cost schedule given an additional global constraint. Let $\Delta = \{S_1, S_2 \cdots S_k\}$ and $\Sigma = \{E_1, E_2 \cdots E_l\}$. For each pair $(S_i, E_j), S_i \in \Delta, E_j \in \Sigma$:

- change the triple of S_i and E_j to (0, 0, 0),
- compute the minimum cost schedule, $S_{i,j}$, and minimum total cost, $C_{i,j}$, as shown in the previous section.

Then a pair (S_i, E_j) which results in the minimum total cost scheduling (i.e., $S_{i,j}$ such that $C_{i,j} = \min_{l,m}\{C_{l,m}\}$) reflects starting and ending objects, which also minimizes the total cost. Note that some fine tuning of the algorithm may reduce the number of iterations by excluding

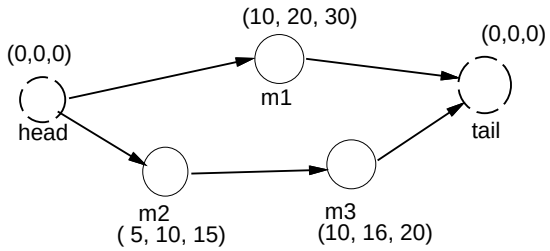

Figure 10:

schedule	m1	m2	m3
	duration(cost)		
1	20 (0)	10 (0)	10 (6)
2	26 (6)	10 (0)	16 (0)
3	22 (2)	8 (2)	14 (2)

Table 2: Three minimal cost schedules

those pairs (S_i, E_j) which cannot be the starting or ending nodes.

The temporal graph in Figure 9 is the one augmented from that in Figure 8 by a global constraint. The minimum and maximum global lengths of the story is 10 and 20 as in Table 1. The author may want to present it in exactly 15 seconds by changing the spring constants to $(15, 15, 15)$. The algorithm produces a minimum total cost solution with pair (n_7, n_9) as the starting and the ending node.

4 Fairness Consideration

We have assumed until now that the presentation quality of an elastic multimedia object is proportional to the distance between its optimal duration and the scheduled duration. A minimum cost schedule of an elastic story globally optimizes the sum of the quality measures of objects by minimizing the total cost as in Equation (3). However, this approach often results in significant differences between the objects in terms of fairness.

Consider a temporal graph of a story with

three objects as shown in Figure 10 and its minimal cost schedules in Table 2. The minimum total cost is 6, and three schedules that incur the same minimum cost are listed. However, in schedule 1 and 2, all of the cost is applied to one object, while in schedule 3 the cost is uniformly distributed across all three objects. We say that schedule 3 is fairer than the other two. The simplex method finds the minimal schedule from one of the extreme points of the convex polyhedral, which is defined by the feasible solution set of the constraints. In doing so, it chooses a set of variables which can be set to zero and tends to generate skewed schedules. The schedules 1 or 2 will be such an example. Intuitively, our goal is to obtain a schedule in which the $cost(I_S)$ is spread across the objects (as in schedule 3) uniformly, subject to the constraints.

We can formulate our problem to minimize the following objective function:

$$cost(Q) = (X_1 - opt_1)^2 + \cdots + (X_n - opt_n)^2$$

With a set of linear constraints and the quadratic objective function, we can solve this minimization problem using the quadratic programming technique [7]. If the constraints are consistent, the technique will find a solution vector, $X(X_1, ..., X_n)$ which minimizes $cost(Q)$. The solution is optimal with respect to the objective function, and provides fairness. Furthermore, the solution is unique if the constraints are satisfiable. But unlike in linear programming, where symbolic computation is sufficient, quadratic programming will involve numerical methods, thus causing computational inefficiency. We propose below approaches of obtaining fairness by iterating the simplex method.

4.1 More Linear Programming on Fairness

Let C_{min} be the minimal cost in (3). By another iteration of the simplex method, we can minimize the cost of the most unfairly scheduled object. In the second iteration, we minimize the new objective function

$$\max_i \{|X_i - opt_i|\}) \tag{4}$$

with an added constraint

$$C_{min} = \sum |X_i - opt_i|. \tag{5}$$

This scheme results in a schedule with minimum worst-case-component among all the possible minimum cost schedules. In the example in Figure 10 and Table 2, the components with largest costs are m_3 (cost 6) in schedule 1 and m_1 (cost 6), in schedule 2. In schedule 3, the worst case cost is 2. Thus the above scheme generates schedule 3.

To minimize the objective function (4) involving the "max" term, we add new constraints

$$|X_i - opt_i| \leq F, \tag{6}$$

for each i, for some F. Minimizing F using simplex method results in a schedule which minimizes (4).

Fairness can also be improved by averaging the skewed schedules generated by the simplex method. In each step of the iteration, we identify a set of objects which have been unfairly scheduled using some measure. We form a new objective function which minimizes the cost of the objects in this set, while preserving the minimum total cost requirement by adding Equation 5 to the set of constraints. A new schedule will reduce the cost of objects in this set, while increasing the costs of other objects which were favored previously. Let $S_1, S_2, ..., S_k$ be the minimal cost schedules thus generated. We get $S = \sum_i a_i * S_i$, where $\sum_i a_i = 1$ and $a_i \geq 0$. This S results in a fairer treatment of each object which has been unfairly treated while preserving the minimal cost. If we average the schedules 1 and 2 in Table 2, the resulting schedule is (23,10,13) with cost (3, 0, 3). This is fairer than schedule 1 and 2, while satisfying the minimum cost requirement.

5 Experiences

The elastic time model has been implemented in Smalltalk/OS2 as part of the Isis authoring environment at the IBM T.J. Watson Research Center. Isis has provided the multimedia substrate for the Home Health-care Prototype System for Children with Leukemia [11, 18], which IBM Research has jointly developed with New England Medical Center. The Isis system is being field-tested currently for building computer-based training systems by non-programmers. The architecture of the system, implementation, and usability issues are beyond the scope of this paper. In this section, we provide a brief discussion summarizing our experiences of working with the system as authors, and of observing authors who are non-programmers. We have learned that the spring system metaphor is an effective one. Authors have found stretching or shrinking media segments such as texts or images most useful.

As for estimating how much time it would take to produce a document, the simplex algorithm is, in a worst case situation, an exponential time method. In practice, however, the algorithm is known to be linear in the number of constraints for most applied problems. Most multimedia stories that our users constructed consisted of less than 20 temporally-related objects. (These were then linked by using user events such as pressing of a button [16], providing a layered model for documents). For all practical purposes, the response time of the system is well within the acceptable range. For instance, the time it takes to schedule a typical 10 object story, including a fairness scheme of optimizing the cost of the most unfairly scheduled object, takes approximately 0.35 – 0.71 seconds on a PS2/Model 95. the variance is due to the differences in the complexities of the constraints. When a global constraint is added, it takes another 0.43 – 0.69 seconds to obtain an alternative schedule. Given a global constraint, the system takes the result of the initial optimal schedule, incrementally adds the global constraint, and computes an alternative.

As the number of temporally-related objects grows arbitrarily and as the complexities of the document increase, an interactive system such as ours can benefit from an incremental method. We are investigating incremental algorithms for building multimedia documents as interactive constraint systems.

6 Conclusions

We have presented the elastic time model for multimedia documents. We have described elastic stories: an elastic story consists of a set of elastic objects and a set of temporal relationships that are defined among the objects. An elastic object is associated with a triple of lengths (spring constants): a minimum, a maximum, and an optimum length. An elastic story (document) also is associated with its corresponding spring constants. As the spring constants for a given object can be changed to reflect changing requirements, the spring constants for an elastic story, which are supplied by the constraint system, can also be modified. As a result, expressive power and flexibility is increased in authoring and browsing the documents.

We have provided a method of obtaining an optimal cost schedule for an elastic story. We have also addressed the issue of fairness in scheduling multimedia stories, and presented approaches to achieve fairness.

The elastic time model presented here may have applicability beyond the multimedia document systems. Any planning system that involves time, and therefore is subject to uncertainties in time, may benefit from an approach such as the proposed elastic time model. It not only provides tolerance to uncertainties, but also provides the users with the capabilities for actively participating in the planning, or the design, process.

Acknowledgement

The authors are grateful to P. Zellweger and an anonymous referee for many useful comments on the draft of this paper.

References

[1] Allen, J., "Maintaining Knowledge about Temporal Intervals", *CACM*, 26(11), 1983.

[2] Buchanan, M. C., Zellweger, P. T. *Automatic Temporal Layout Mechanisms.* ACM Multimedia, 1993.

[3] Buchanan, M. C., Zellweger, P. T. *Automatically Generating Consistent Schedules for Multimedia Documents.* Multimedia Systems Journal, 1(2), Springer-Verlag, 1993.

[4] Dechter, R., Meiri, I., and Pearl, J., "Temporal Constraint Networks". *Artificial Intelligence*, 19: pp61-95, 1991.

[5] Diaz, M., Senac, P. *Time Stream Petri Nets: A Model for Timed Multimedia Information.* Proc. 15th Conf. on Application and Theory of Petri-nets, 1994.

[6] Fiume, E. et al. *A Temporal Scripting Language for Object-Oriented Animation.* Eurographics, 1987.

[7] Glass, S. *Linear Programming Methods and Applications.* McGraw-Hill Book Company, pp. 315, 1975.

[8] Hamagawa, R., Sakagami, H., Rekimoto, J. *Audio and Video Extensions to Graphical User Interface Toolkits.* ACM Multimedia, 1993.

[9] Little, T. and Ghafoor, A. *Synchronization and Storage Models for Multimedia Objects.* IEEE Journal on Selected Areas in Communications, 8(3), 1990.

[10] Kim, M. Y., Song, J. *Hyperstories: Combining Time, Space, and Asynchrony in Multimedia Documents.* IBM Research Report RC 19277, 1993.

[11] Kim, M. Y. *Guardian: A Knowledge-Based Home Health-Care Support System for Children with Leukemia.* IBM Research Report RC 19858, 1994.

[12] Kim, M. Y. *Creative Multimedia for Children.* Conference on Human Factors in Computing Systems, 1995.

[13] Levin, R., Kirkpatrick, C. *Planning and Control with PERT/CPM.* McGraw-Hill, 1966.

[14] MacNeil, R. *Generating Multimedia Presentations Automatically using TYRO, the Constraint, Case-Based Designer's Apprentice* IEEE Workshop on Visual Languages, 1991.

[15] Press, W., Flannery, B., Teukolsky, S., Vetterling, W. *Numeric Recipies in C: The Art of Scientific Computing.* Cambridge University Press, 1988.

[16] Song, J., Doganata, Y., Kim, M. Y., Tantawi, A. *Modeling Timed User Interactions in Multimedia Documents.* IBM Research Report RC 19933. 1995.

[17] Stotts, P.D., Furuta, R. *Temporal Hyperprogramming.* Journal of Visual Languages and Computing, vol. 1, 1990.

[18] Tetzlaff, L., Kim, M. Y., Schloss, R. *Home Health-Care Support System for Children with Leukemia.* Conference on Human Factors in Computing Systems, 1995.

[19] van Rossum, G. et al. *CMIFed: A Presentation Environment for Portable Hypermedia Documents.* ACM Multimedia, 1993.