# Time and Uncertainty in Reasoning about Order

**Robert A. Morris** and **Dan Tamir**
Florida Institute of Technology
Melbourne, FL 32901
email:morris@cs.fit.edu

**Kenneth M. Ford**
University of West Florida
Pensacola FL 32514
email: kford@ai.uwf.edu

**Abstract**

The ability to intelligently order events is important for planning and scheduling in the presence of uncertainty about the expected duration of those events. This paper presents a time-based theory of an agent in a dynamic environment, and a framework for reasoning for the purpose of generating effective orderings of events.

## 1 Setting the Stage

In this study, the interest is in the role of time in the ability of intelligent agents to plan or schedule events, especially actions, events of which they are the agent. Researchers in AI have, for a number of years, offered analyses and computational models of the temporal reasoning underlying these abilities. These models have explained the intuitive complexity of reasoning about time in terms of proving the consistency of a set of temporal constraints, an inherently intractable problem. This study adds further complexity by folding into the framework a dynamically changing environment, wherein temporal knowledge becomes outdated, as well as being partial and incomplete. How, we ask, can an agent utilize the information found in such an environment in order to effectively solve planning and scheduling problems?

The impetus for this investigation is a system which provides a solution to the difficult problem of scheduling telescope observations [1]. This solution required attention be given to the fact that the duration of a repeating event may be different on different occasions. This made any generated schedule "fragile", which means that there was a tendency for it to "break" during execution. The novelty of the approach of the researchers was the integration of statistical information about past occurrences of events in order to predict how well a schedule will stand up against a contrary night sky. This allowed for sensitive locations in the schedule to be identified, and made it feasible to maintain a library of contingency schedules. We feel this approach to solving planning and scheduling problems in a changing world can be extended and generalized to other problems with similar, dynamic environments. One objective of this study is to perform these transformations.

The objectives of this paper consist of

- Constructing an abstract representation of the intelligent behavior which is manifested in the telescope scheduling example, as well as others;
- Proposing a formal representation of the knowledge required to realize this behavior; and
- Presenting a computational model of learning the requisite knowledge based on statistical evidence inferred from the experience of temporal duration and order.

## 2 Abstract Representation of Behavior

The interest here is in systems embedded in a dynamic environment with feedback in the form of rewards. It is desirable for the system to learn from these rewards in order to maximize its rewards over the long run. Traditionally, such a system is modeled in terms of *state transition networks*, consisting of states, actions and state-transition functions. Here, an alternative model is presented with an underlying temporal ontology. Specifically, there are events represented by their *durations*, and a single atomic temporal ordering relation, *immediately precedes* ($<$).

Given a set $E = \{A, B, C\}$ of events, if an agent prefers a certain ordering of their occurrences, say, $A < B < C$ ("$A$ immediately before $B$ immediately before $C$") to another, the reason may have to do with constraints which lead to a preference for that order. There are many varieties of constraints possibly underlying this preference. Here the interest is in criteria for orderings that are based on *temporal constraints*. One example of such a constraint involves minimizing the *overall extent* of the performance of all the tasks. By overall extent is meant the interval of time it takes all the events in $E$ to complete. On this criterion, an ordering of $E$ which is expected to minimize the overall extent of $E$ will be the most preferred ordering.

Another criterion for ordering will be in terms of minimizing the overall *duration uncertainty* of the set of tasks. Intuitively, duration uncertainty is manifested in terms of a relative lack of confidence concerning how long an event, or a set of events, will take. If it is possible to predict that one ordering of the tasks will exhibit less duration uncertainty than another, then choosing the ordering with less uncertainty will be preferred. This is analogous to taking a "sure bet", even if the payoff is less than another choice which is less likely.

The inability to predict how long an event will last on a given occasion (duration uncertainty) is a pervasive feature of common sense experience. Things that happen in a given day, e.g., breakfast, driving to work, faculty meetings, going to the dentist, exhibit varying amounts of duration uncertainty. Duration uncertainty is undesirable to a rational agent because it leads to failure in the completion of plans and schedules, and the need for time-consuming repair and revision.

To satisfy one or the other of these constraints, an agent can choose to order the occurrences of the events in such a way that events in close temporal proximity share one or more *stages*. Informally, a stage of an action or event $E$ is an action or event which occurs as part of the occurrence of $E$. For example, "preparing the cleaning utensils" can be viewed as a stage in most or all cleaning actions. Often, an event can be "sliced" in different ways to uncover its stages.

Suppose two cleaning room actions, *clean kitchen* ($K$) and *clean bath* ($B$) are performed together, say $K < B$. There will be a tendency for the preparation stage of $B$ to not be required (or be simplified); hence the overall duration of performing both should be reduced. Furthermore, since the duration uncertainty of the whole will be a function of the duration uncertainty of the different stages, there's a chance that duration uncertainty can also be reduced as a result of this pairing. This situation is illustrated in Figure 1. In this figure, stage $S_1$ of $K$ is shared with $B$. The temporal effect of sharing stages is that the events can be viewed as overlapping in time. Notice that when speaking of such relations, there is no assumption of *convexity* (no interruption) with respect to the intervals making up the durations of the events. It follows that an agent should be able to more accurately predict how long the bathroom cleaning will take *when preceded by the*

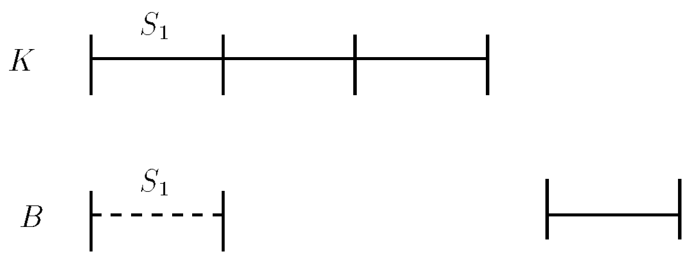

Figure 1: Effect of Pairing Similar Events in Close Temporal Proximity

*kitchen cleaning action* than it could predict its duration in isolation, or when preceded by a event sharing no stages with it.

The point of the examples, then, is that *events that share stages will tend to be mutually influencing with respect to duration, especially when paired in close temporal proximity.* This sort of information would be useful for an agent who is either lacking the requisite knowledge about the events for which it needs to find an intelligent ordering, or in which the environment is constantly changing, making its knowledge outdated. Consider, for example, a robot assigned the task of delivering mail in a dynamically changing environment. Offices may move, for example, or construction to different parts of the complex may require dynamically revising the routes, and hence possibly the order, in which mail is delivered. Similarly, it may be equipped with only a crude or outdated map of its environment.

We proceed to formalize a model of an agent in a dynamically changing environment. The model is based on the familiar idea of using a network to store temporal information. Here, the nodes, or variables represent events in terms of their durations, and the arcs store values which represent the effect of orderings of events on the durations of events that follow them in close temporal proximity.

**Definition 1** *(Duration Network )*A **Duration Network** $N$ is a set of variables $V = V_1, \ldots, V_n$, and a set of labeled edges $E = \{\langle V_i, V_j \rangle : \forall V_i, V_j \in V\}$. An **instantiation** of $N$ is a function $I : V \cup E \rightarrow Z$, such that, for all $V_i, V_j \in V$, $E_{i,j} \in E$:

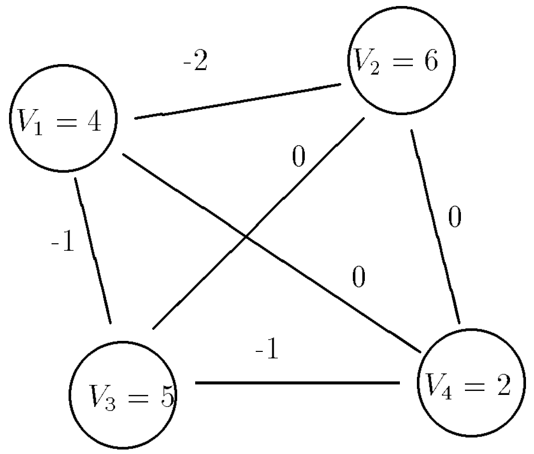

Figure 2: Instantiated Duration Network

- $I(V_i) > 0$;
- $I(E_{i,j}) \leq 0$; and
- Let $E_{i,j} = \langle V_i, V_j \rangle$. Then $|I(E_{i,j})| < I(V_i)$ and $|I(E_{i,j})| < I(V_j)$.

A duration network is a complete network in which the variables stand for events, and their values are durations of these events. The labels on the arcs represent the effect of sharing stages on durations. A negative value for $I(\langle V_i, V_j \rangle)$ represents the advantage of performing $V_i$ and $V_j$ together by virtue of their sharing a stage; the negative value is the "reward" for doing them in close temporal proximity. Figure 2 depicts an instantiated duration graph. To illustrate the meaning of the graph, consider the nodes $V_1$ and $V_2$. The order $V_1 < V_2 < V_3$ would yield a "reward" of 2 time units. This means that the overall duration of performing this sequence would be $4 - 2 + 6 + 0 + 5 = 13$ time units. Compared with performing $V_1 < V_3 < V_2$, which has overall duration $4 - 1 + 5 + 0 + 6 = 14$ time units, the first ordering would have the smaller overall extent.

It is useful to distinguish what we will call *legs* of a tour of a duration network $N$. Intuitively, if a tour is a complete path through the network, a leg of the tour is any subpath of that path. More formally, we use the notion of *sub-sequence* of a sequence (using the notation $t' \sqsubseteq t$) to characterize tour legs. If $t = \langle V_{t_1}, V_{t_2}, \ldots V_{t_n} \rangle$ is a tour through $N$, then, for example, $\langle V_{t_3}, V_{t_4}, V_{t_5} \rangle$ is a leg. To relate a leg to its tour, we use $t/\langle V_{t_i}, \ldots V_{t_i+m} \rangle$ to mean "the part of $t$ consisting of the indicated leg".

**Definition 2** *(Process/Tour of a Duration Network)*A **k-process** of a duration network $N = (V, E)$ is a sequence $P = \langle I_1, I_2, \ldots, I_k \rangle$ of instantiations of $N$. A **tour** $t$ of a duration network $N$ with variables $V = \{V_1, \ldots, V_n\}$ is a permutation of $V$. We write $t = \langle V_{t_1}, \ldots, V_{t_n} \rangle$ to enumerate the elements of $t$.

**Definition 3** *(Cost of a Tour/Tour Series)*Given an Instantiation $I$ and tour $t = \langle V_{t_1}, \ldots, V_{t_n} \rangle$, the cost of a tour in $I$ ($c(t, I)$) is

$$c(t, I) =$$

$$I(V_{t_1}) + I(\langle V_{t_1}, V_{t_2} \rangle) + \ldots + I(\langle V_{t_{n-1}}, V_{t_n} \rangle) + I(V_{t_n})$$

Given a $k$-process $P = \langle I_1, \ldots I_k \rangle$ and a sequence of corresponding tours $T = \langle t_1, \ldots, t_k \rangle$, called a **tour series**, the cost of the series $T$ in process $P$ ($C(T, P)$) is

$$C(T, P) = \sum_{1 \leq i \leq k} c(t_i, I_i)$$

More generally, we can introduce the notion of "cost of a leg $L = \langle V_{t_i}, \ldots V_{t_{i+m}} \rangle$ of a tour $t$ in $I$" as follows:

$$c(t/L, I) = \begin{cases} 0 & : \; L \not\sqsubseteq t \\ c(L, I) & : \; otherwise \end{cases}$$

Finally, we can speak of the cost of a leg in a tour series $T$ and a process $P$:

$$C(T/\langle V_{t_i}, \ldots V_{t_{i+m}} \rangle, P)$$

as the sum of the costs of this leg in all the tours in the series containing this leg.

The interest now is to define a set of one-person games involving tours of the duration graph. The specific goal of interest is to find a tour series $T_{mest}$ of length $k$ which is an agent's estimate of the **minimal tour series** $T_{min}$. The latter is the tour series which, given a duration network $N$ and $k$-process $P$, incurs the minimum cost over all possible tour series.

Other goals are of course possible. One is to minimize the standard deviation from the mean of tour durations in the series. Another goal is to complete as many of the events (i.e., visit as many of the variables) as possible, given rigorous time constraints (i.e. cost). Other versions of the game differ on assumptions concerning either the agent's initial knowledge of $N$, its abilities to update the knowledge based on experience in the form of tours it has made, or on the properties of $P$. The interest is in finding definitions of $P$ which characterize properties and relations of the abstract world which are homomorphic to those properties and relations which occur in real world planning and scheduling domains.

First, let us say that an instantiation $I$ *is totally repeating* in $P = \langle I_1, I_2, \ldots I_m \rangle$ if $\exists i, j i \neq j, I = I_i = I_j \in P$. We can refine this to "repeats $n$ times" in an obvious way. Two total repetitions of $I$, say $I_m$ and $I_p$ are *$n$ units apart* if $\|m - p\| = n$. If $n = 1$, then the repetitions will be said to be consecutive. $I$ will be said to be *p-periodic* in $P$ if any pair of occurrences of $I$ in $P$ repeat $r$ units apart, where $r$ is a factor of $p$. Similarly, $I$ is *almost periodic* in $P$ if there exists a $p$ such that any pair of occurrences of $I$ in $P$ occur a distance apart which is "close" to being a factor of $p$. We assume this notion of being almost periodic is intuitive enough to remain qualitative, although obviously it can be made more precise. Finally, we can define a notion of a *partially repeating* instantiation in $P$, and derivative notions, in terms of instantiations that share some of their values.

Secondly, a process will be said to be *invariant* if the values of the different instantiations do not differ a great deal. We distinguish two kinds of invariance, *duration* and *path* invariance. First, consider total duration invariance. We can draw an even finer distinction between *weak* and *strong* total duration invariance. We can express strong invariance in terms of mean, or average duration, and standard deviation. Thus, let the mean duration of an event represented by $V_i$ in a process $P$ be the average duration of $V_i$ over all instantiations in $P$. Let $\sigma^P_{V_i}$ be a variable denoting the standard deviation from the mean. We say that a process $P$ is *$\mu$-invariant* if for each $V_i$, the value of $\sigma^P_{V_i}$ is less than $\mu$. Finally, we say that a process $P$ is *totally invariant* if there exists a $\mu$ which is close to 0 such that $P$ is $\mu$-invariant.

Path invariance means that there is never a large difference in the cost among different paths through $N$ throughout a process $P$. More precisely, let $c(t_1, I)$ and $c(t_2, I)$ be the costs of any two of the $n!$ tours through a duration network $N$ with $n$ variables, given $I$. Then path invariance implies that the the difference between these values is not greater than some small value $\epsilon$.

Strong invariance is a global property of a process: intuitively, it says that the duration of any variable or edge of a duration network never strays excessively from the mean. This does not allow a "real good" path ever to become "real bad", although it may become less good. Weak invariance is a strictly local phenomenon: it constrains every pair $I_i, I_{i+1}$ of consecutive instantiations in $P$ to be "close in their assignments" to all elements of $N$. (This notion can be made precise in an obvious manner.) Thus, weak invariance allows for a good path to become bad over the long run. We can generalize any of these notion of invariance to (partial) invariance, in which a subset of $I$ exhibits invariance. Again, for our purposes, it is enough to leave this intuitive notion qualitative.

We view the world as exhibiting varying degrees of invariance and periodicity. An intelligent agent can learn and apply knowledge about invariance and periodicity in order to make plans which are intelligent. This is the case although the knowledge the agent has is incomplete, and partial, and the world is in constant flux. In the next section, we consider this capability in the context of constructing the tour series $T_{mest}$.

## 3 Computational Theory

As noted, the ability of an agent to effectively solve the class of problems abstractly characterized as a traversal of a duration network depends on

- The goal of the game;
- The properties of $P$; and
- Assumptions about the agent's knowledge of $N$ and $P$.

For example, if the agent is given the requisite knowledge to determine $P$, then it does not matter whether $P$ exhibits any invariance or periodicity: the agent will be able to "precompute" an optimal $T_{mest}$ based on an exhaustive search of each instantiation.

The case to be examined here is the one in which the knowledge the agent has of $P$ is, at best, partial. In this section, we describe a version of the game in which

1. The agent has, initially, an "abstract map" of $N$;
2. The agent has no quantitative knowledge about $P$;
3. $P$ exhibits total strong duration and path invariance.

4. The goal of the game is for the agent to construct $T_{mest}$.

We next present a computational theory which explains and realizes this behavior.

To solve for a goal, given the initial constraints, the agent needs to have a means to learn and apply knowledge it discovers about $P$ to select a tour $t_j$, given $t_1, \ldots, t_{j-1}$, as part of a series. To make use of the rewards afforded by certain paths, we introduce the notion of relative mean duration:

**Definition 4** *(Relative Duration)* Let $N = (V, E)$ be a duration network, $T = \langle t_1, \ldots t_k \rangle$ be a tour series and $P = \langle I_1, \ldots, I_k \rangle$ be a process. The *relative duration* of an event $V_i$ with respect to an event $V_j$ in an instantiation $I_n$ and corresponding tour $t_n$ ($rd(\langle V_i, V_j \rangle, t_n, I_n)$) is $c(t_n / \langle V_i, V_j \rangle, I_n) + c(t_n / \langle V_j, V_i \rangle, I_n)$. Furthermore, the *relative mean duration* of an event $V_i$ with respect to an event $V_j$ over a set of $k$ occurrences of $V_i$ and $V_j$ ($rmd_{V_i,V_j}(I, T)$)is

$$\frac{C(T/\langle V_i, V_j \rangle, I) + C(T/\langle V_j, V_i \rangle, I)}{k}$$

Let $\sigma_{rmd_{V_i,V_j}}(I, T)$ denote the standard deviation of $rmd_{V_i,V_j}(I, T)$. If $I$ and $T$ are given, the notation for these values is simplified to $rmd_{V_i,V_j}$ and $\sigma_{rmd_{V_i,V_j}}$.

Intuitively, relative duration is the cost of the leg $V_i < V_j$ or $V_j < V_i$ in a tour, given an instantiation of the variables and the edge connecting them. Since the relation of "sharing a stage" is symmetrical, these costs are assumed to be identical; e.g., any reward for pairing cleaning actions $K$ and $B$ in immediate temporal proximity will be collected, whether the order be $K < B$ or $B < K$. Relative mean duration, then, consists of the average relative duration of pairs of events over a set of tours in a series.

Assuming $P$ exhibits total duration and path invariance, an agent can incrementally learn the requisite knowledge for constructing $T_{mest}$ on the basis of computing and storing relative mean durations. This information will be stored in what will be called a $\sigma$-network:

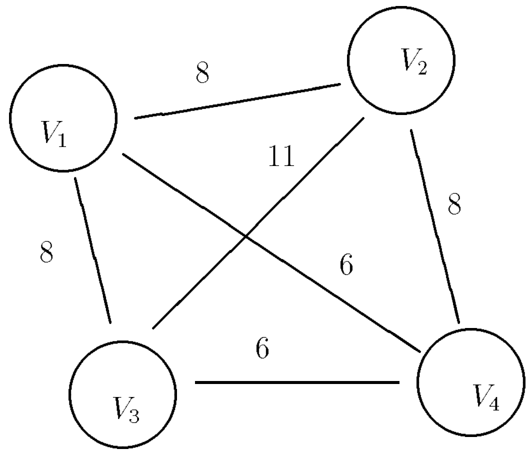

Figure 3: A Possible $\sigma$-Graph Associated with Figure 2

**Definition 5** Given a duration network $N$ and a process $P$, a $\sigma$-network for $N = (V, E)$ is a weighted undirected network with the following characteristics. Each vertex is labeled by one of the elements in a set $V$. Each edge $(V_i, V_j)$ is labeled. The value of the label represents $rmd_{V_i,V_j}$.

Figure 3 is an example of a $\sigma$-network corresponding to the duration network in the previous figure. The labels on each edge represent values for $rmd_{V_i,V_j}$. These values would be accurate, for example, at the end of a $k$-process $P$ consisting of $k$ repetitions of the instantiation depicted in the previous figure. With the information in the $\sigma$-network, an agent can determine the next best tour in $T_{mest}$.

Let us assume that the process $P$ exhibits strong duration and path invariance, but incorporates no assumptions about periodicity. The method TS for constructing $T_{min}$ is summarized in Figure 4. For the sake of simplicity, there is an assumption of a "learn-

**Algorithm** TS

*Input*:

- A process $P = \langle I_1, \ldots, I_k \rangle$;
- A Duration Network $N = (V, E)$, $V = \{V_1, V_2, \ldots, V_n\}$, initialized by an instantiation $I_{init}$;
- A $\sigma$-network $\Sigma = (V_\sigma, E_\sigma)$, where $V_\sigma = V$ and $E_\sigma = E$. For each edge $E_{i,j}$ in $\Sigma$, let $v(E_{i,j})$ represent the value of the label on that edge. Initially, this value is 0.

*Output*:

- The updated $\sigma$-network $\Sigma$, which now contains statistical information about $P$ based on its having executed a tour series $T_{mest} = \langle t_{mest_1}, \ldots t_{mest_k} \rangle$; and
- $C(T_{mest}, I)$.

```
For each edge $E_{i,j}$ in $\Sigma$ do
    $v(E_{i,j}) \leftarrow rd(\langle V_i, V_j \rangle, I_{init})$
    $k \leftarrow 1$;
$T_{mest} \leftarrow \langle\rangle$
    loop
    $t_k \leftarrow HamiltonPath(\Sigma)$
    $UpdateMean(\Sigma, t_k, I_k, k)$
    $T_{mest} \leftarrow T_{mest} + \langle t_k \rangle$
/* $\langle u \rangle + \langle v \rangle = \langle u, v \rangle$ */
    $k \leftarrow k + 1$
    until $k = p$
Return $\Sigma$ and $C(T_{mest}, P)$
```

Figure 4: Algorithm for Constructing a Tour Series which Minimizes Overall Extent

```
UpdateMean($var\Sigma : \sigma - graph, t : tour, I : instantiation, k : index$)

begin
For each edge $E_{i,j} = \langle V_i, V_j \rangle$ in $\Sigma$ do
    $rmd_{V_i,V_j} \leftarrow c(t_k / \langle V_i, V_j \rangle, I) + c(t_k / \langle V_j, V_i \rangle, I)$
    if $rmd_{V_i,V_j} > 0$ then
        $v(E_{i,j}) \leftarrow \frac{(k-1)[v(E_{i,j})] + rmd_{V_i,V_j}}{k}$
end
```

Figure 5: Updating Algorithm for $\Sigma$-graph

ing phase" in which the agent is supplied values for one instantiation $I_{init}$ of $N$. This can be viewed as, e.g., a robot being supplied a "map" of the world it needs to navigate repeatedly. The main loop iteratively generates $t_k$, the next tour in the series, from $\Sigma$ by performing a Hamilton Tour of this network, and updates $\Sigma$ based on information it has acquired about $I_k$ as the result of its tour $t_k$. The Hamilton tour gives the best estimate of the tour with the lowest overall extent. The "final score" of the game is the overall cost of the tour series $T_{mest}$.

The UpdateMean Algorithm records the cost of $t_k$ as the result of $I_k$, by updating the $\Sigma$-network accordingly. The procedure is summarized in Figure 5. This procedure simply updates the mean relative duration $rmd_{V_i,V_j}$ for each edge in $\Sigma$, based on the result of the cost of traversing this edge in $I_k$ by tour $t_k$ (if this tour contains this leg; if not, then this cost is 0 and no updates are made). This algorithm, we claim, realizes behavior which, under the constraints posed by this version of the game, is useful in the generation of intelligent orderings of a set of events.

As a variation on the game, suppose the agent is interested, not in reducing overall extent, but rather in reducing the duration uncertainty associated with a set of events. This would be the case, e.g., if the agent has no constraints on the time of the completion of a set of tasks, but wanted to be reasonably sure, at each moment in every tour, on which leg of the tour it is located. A minor modification of the game in the preceding section will allow the agent to estimate a tour series $T_{du}$, which approximates the tour series which is minimal with respect to duration uncertainty.

Again, let $\sigma_{rmd_{V_i,V_j}}$ be the standard deviation of the relative mean duration of a set of occurrences of events $V_i$ and $V_j$ in immediate temporal succession. Imagine modifying the $\sigma$-network so that the labels on each edge $E_{i,j}$ stores values of $\sigma_{rmd_{V_i,V_j}}$. We can replace UpdateMean with a procedure, call it UpdateSD, for updating standard deviations, based again on the result of the most recent tour. Then, applying TS with UpdateSD computes $T_{du}$, which estimates the tour series with the minimal duration uncertainty.

Numerous other enhancements to the representation are possible. For example, incorporating duration uncertainty as a constraint would lead to a variation of the one-person game in which the agent's goal is to minimize the duration uncertainty associated with a tour. Another enhancement to the game involves incorporating assumptions regarding periodicity to $I$ would make such information useful to store in a $\sigma$-network. Relative durations would be further relativized to time periods, which are represented by the index $k$ on the instantiation $I_k$. Relative mean durations, and their standard deviations, would be required to reflect this relativization. For this information, it is possible that a quantitative model for probabilistic temporal reasoning such as found in [2], could be applied; alternatively, a qualitative model of recurrence, such as [3], might serve the same purpose.

## 4 Conclusion

This paper has provided a framework for developing planning and scheduling systems in a dynamic world. One primary assumption motivating this framework is that events tend to exhibit varying degrees of duration uncertainty, and that an intelligent agent needs to confront this uncertainty in planning situations. One aid in reducing duration uncertainty exploits the fact that events share stages with other events.

## References

[1] Drummond, M.; Bresina, J.; Swanson, K., 1994. Just-In-Case Scheduling. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94).* AAAI Press/MIT Press, Menlo Park, 1994:1098-1104.

[2] Goodwin, Scott D., Hamilton, H. J., Neufeld, E., Sattar, A., Trudel, A. Belief Revision in a Discrete Temporal Probability-Logic. *Proceedings of Workshop on Temporal Reasoning, FLAIRS-94,*

[3] Morris, R., Shoaff, W., and Al-Khatib, L., (1994) Domain Independent Reasoning About Recurring Events. Forthcoming in *The Journal of Computational Intelligence.*