Clothespins on Timelines: Utilities and The Interval Representation of Time

Authors: R. P. Loui and Jersey Chen

We discuss the problem of representing utility in planning systems that are based on Allen's [83] popular ontology for planning, which represents actions and events as time intervals. We identify a small number of primitive functions on time intervals which may be helpful in representing preference and also in eliminating dominated actions. Assuming that utility can be decomposed to take advantage of these primitives, these functions provide one solution to the problem of specifying utility in such expressive planning languages. We identify a restricted class of utility expressions that generate linear programming problems. The contribution is not deep, but is instructive. We conclude with the pessimistic observation that any extension to the Allen framework to support DMUR destroys much of the initial appeal of the system. What remains is the ontological emphasis on intervals. It may yet be appealing to some who find the ontology cognitively concordant, but it forces the introduction of a metric that Allen had originally sought to avoid.

... Read complete abstract on page 2.
Clothespins on Timelines: Utilities and The Interval Representation of Time

Complete Abstract:

We discuss the problem of representing utility in planning systems that are based on Allen's [83] popular ontology for planning, which represents actions and events as time intervals. We identify a small number of primitive functions on time intervals which may be helpful in representing preference and also in eliminating dominated actions. Assuming that utility can be decomposed to take advantage of these primitives, these functions provide one solution to the problem of specifying utility in such expressive planning languages. We identify a restricted class of utility expressions that generate linear programming problems. The contribution is not deep, but is instructive. We conclude with the pessimistic observation that any extension to the Allen framework to support DMUR destroys much of the initial appeal of the system. What remains is the ontological emphasis on intervals. It may yet be appealing to some who find the ontology cognitively concordant, but it forces the introduction of a metric that Allen had originally sought to avoid.

This technical report is available at Washington University Open Scholarship: http://openscholarship.wustl.edu/cse_research/327
Clothespins on Timelines: Utilities and The Interval Representation of Time

R. P. Loui and Jersey Chen

WUCS-93-05

February 1993

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis MO 63130-4899

This work was supported under NSF R9008012 (Loui) and NSF CDA9102090 (Chen).
Clothespins on Timelines:
Utilities and The Interval Representation of Time

R. P. Loui
Department of Computer Science
Washington University
St. Louis, MO 63130
loui@ai.wustl.edu

Jersey Chen
Harvard University
Cambridge, MA 02138
jersey@ai.wustl.edu

Abstract.
We discuss the problem of representing utility in planning systems that are based on Allen’s [83] popular ontology for planning, which represents actions and events as time intervals.

We identify a small number of primitive functions on time intervals which may be helpful in representing preference and also in eliminating dominated actions. Assuming that utility can be decomposed to take advantage of these primitives, these functions provide one solution to the problem of specifying utility in such expressive planning languages. We identify a restricted class of utility expressions that generate linear programming problems. The contribution is not deep, but is instructive.

We conclude with the pessimistic observation that any extension to the Allen framework to support DMUR destroys much of the initial appeal of the system. What remains is the ontological emphasis on intervals. It may yet be appealing to some who find the ontology cognitively concordant, but it forces the introduction of a metric that Allen had originally sought to avoid.

Acknowledgements.
This work was supported under NSF R9008012 (Loui) and NSF CDA9102090 (Chen). The second author participated while an undergraduate, concentrating in economics. Thanks to Noémi Berry for the planning example.
Problem.

James Allen’s advance for planning ten years ago [83] was to introduce an ontology for actions, events, and properties that made little distinction between the three classes. All were time intervals, and the main distinction was that actions were schedulable, events were not, and properties were implied by actions or by events, or by interactions. Furthermore, thirteen primitive predicates could be used in a language that described relations among the intervals, such as two intervals overlapping, or one interval being before another. These predicates permitted aesthetically pleasing and epistemologically honest imprecision about when events or actions actually occurred, with respect to reference times (e.g., 8pm). Knowledge representers were not obliged to refer to time points; so they did not, for example, have to say whether a light was on or off at the moment at which it was switched. More importantly in practice, the presumption that actions took time and could be simultaneous in a variety of ways was built into the ontology; discrete sequential action and event was a very special case. Finally, inference and update could be done using a simple constraint-propagation scheme.

Allen’s scheme has been popular in the planning community, although its adherents readily admit that the use of time points for knowledge representation (e.g., McDermott [82]) would also have worked. Its popularity may be due to the willingness of planning people to think in terms of temporal intervals rather than points, when the temporal aspects of the planning problem become complex. Certainly Allen’s ontology is a useful way of representing non-linear plans.

The problem is how to be faithful to Allen’s time-interval ontology, while extending it to permit DMUR (decision-making under risk), or equivalently, to permit soft goal constraints. In particular, there need to be ways of specifying probability and utility.

It is hard enough to extend AI’s state-based planning model (e.g., STRIPS or situation calculus) to accept probabilities and utilities (see the work by Breese [87], Wellman [88], Loui [88,89,90], Haddawy–Hanks [90], Haddawy [91], Hanks [91], and Breese–Goldman–Wellman [91]). The main problem with extending state-based planners for DMUR is that logical language is used to represent states of the world. n states of the world would require n utility values, and a propositional language of k propositions makes n = 2^k. It is much worse for predicate-logical languages, and another exponential worse when heuristic utility is used, so that values must be given for sets of worlds when expectations cannot be taken (this is discussed in Loui [88]). Of course, all \( \exp(2, \exp(2, k)) \) numbers could be given quite easily if there were a regularity to the preference, such as all states having equal utility.

The situation is worse with temporal reasoners. When the relevant conception of the world is a history (or scenario) rather than a snapshot, the distinguishable objects of value proliferate remarkably. Suppose utility measures the number of times I have held ice cream in my hands over the past ten years, instead of just measuring my present state of ice-cream-holding; then outcomes are histories, and there are numerous histories. For one thing, distinguishable histories may lead to the same resultant “state.”

This interest in histories could have happened with state-based planners, except that the idea of state in the first place is that state is somehow complete. If ice-cream countings were relevant to preference, accumulations could have been included in state. If there were many such relevant
accumulations, the description of state would grow. With enough accumulations, it might be easier to make utility a function of world history rather than world state. But state-based conceptions bias against this way of thinking. Meanwhile, overlapping time intervals that describe scenarios of world devolvement encourage the opposite thinking.

The Savage ontology for DMUR [50] does not require the use of state instead of history. But most users of DMUR implicitly conceive of the world as a succession of states, with value determinable from some final state. In any case, to our knowledge no one has investigated how to express utility in planning systems that use Allen’s thirteen temporal relations, and ontology of temporal intervals, for expressing knowledge about the world.

Technical Preliminaries.

Figure 1 shows the simplest example of knowledge in the temporal language, first as a set of constraints, then as this set’s depiction.

\[ o(Make-Pasta, Eat-Pasta) \land \\
\langle(Make-Pasta, Guests-Leave) \land \\
\langle(Eat-Pasta, Guests-Leave) \]

![Diagram of temporal intervals]

Figure 1. The schedulable action, Make-Pasta, overlaps the schedulable action, Eat-Pasta, and both are before the event Guests-Leave. Actions are in bold.

The possible temporal relations are: before (\(<\)), equal (\(=\)), meets (m), overlaps (o), during (d), starts (s), finishes (f), and their inverses (e.g., after (\(>\)), is–met–by (mi), is–overlapped–by (oi), contains (di), is–started by (si), and is–finished–by (fi); equal is its own inverse. Note that the times of the intervals are only constrained to satisfy the stated relations; the knowledge is ambiguous between all of the various ways those constraints can be met (that are distinguishable if reference times, such as 8pm, are used). The language is inherently vague; it lends itself to abstraction.

The effects of scheduling and events occurring are recorded by differences in scenario, where each scenario is a set of temporal relations. For example, figure 2 shows the scenario for a different plan, where the two schedulable actions are non–simultaneous, and where the occurrence of the event is less convenient.

Which scenario is preferred? We need a function to convert each description of outcome into its real–valued utility. Clearly some kind of decomposition of this function is needed, in the same
way that multi–attribute utility forces a linear decomposition and thereby allows combinatorial gains in simplification.

Our observation is that the Allen–style vagueness is especially useful in expressing utility. Actual scenario development frequently can refer to reference times; in fact, depending on how probability information is represented, it may have to refer to reference times in order to generate a scenario of measurable likelihood. It appears to be the specification of goals that benefitted from the ability to specify that two time intervals overlapped, regardless of what was the absolute time they overlapped.

Our solution is to provide the following additional primitive relations on time intervals $A, B$:

- $u(+)(tod)(A,B)$ utility is increasing in the duration of $A$ within $B$;
- $u(+)(tov)(A,B)$ utility is increasing in the overlap of $A$ upon $B$;
- $u(+)(tbm)(A,B)$ utility is increasing in the time between $A$ meeting $B$;
- $u(+)(tbs)(A,B)$ utility is increasing in the time between $A$ starting $B$;
- $u(+)(tbf)(A,B)$ utility is increasing in the time between $A$ finishing $B$;

and corresponding relations for decreasing utility:

- $u(-)(tod)(A,B)$ utility is decreasing in the duration of $A$ within $B$;
- $u(-)(tov)(A,B)$ utility is decreasing in the overlap of $A$ upon $B$;
- $u(-)(tbm)(A,B)$ utility is decreasing in the time between $A$ meeting $B$;
- $u(-)(tbs)(A,B)$ utility is decreasing in the time between $A$ starting $B$;
- $u(-)(tbf)(A,B)$ utility is decreasing in the time between $A$ finishing $B$.

All are considered to be directional, so one might reasonably assert

$u(+)(tod)(A,B) \& u(+)(tod)(B,A)$,

which asserts that utility accrues in the amount of $A$'s during $B$ or vice versa, if either occurs. Just as in Allen, we define inverses, $todl, tovi, tbml, tbsl,$ and $tbfj,$ and we write lists of relations to indicate disjunction (so write $u(+)(tod todl)(A, B)$).
Each relation presumes that a certain condition (a guarding condition) holds before the relation has relevance to utility. These are:

- \( t_{od} \) presumes that \( A (d) B \)
- \( t_{ov} \) presumes that \( A (o) B \)
- \( t_{bm} \) presumes that \( A (<) B \)
- \( t_{bs} \) presumes that \( A (< o) B \)
- \( t_{bf} \) presumes that \( A (< d) B \).

The idea is to have a collection of relations between time intervals that contribute to utility. Their contribution is not just by holding as a condition, thereby defining a goal condition, but by contributing to utility monotonically in the size of some interval, e.g., the interval of overlap, if such an interval exists, e.g., if there is any overlap. Each relation (or each set of relations between any two time intervals) we call a value item. Presently we do not have the information to combine the contributions of value items.

We do not claim that this set is complete in any sense, and there are clear additions and extensions. In fact, we will soon augment this language so that tradeoffs will be possible. However, the idea of expressing utility in this form has several immediate advantages:

- it is within the time interval paradigm
- it permits useful decomposition
- it identifies a most useable class of objectives
- it allows some metareasoning about dominated actions: namely, extending or shortening unconstrained intervals in which utility is respectively increasing or decreasing.

**Extended Example.**

Consider the example in figure 3. The problem is scheduling the time of a flight from SJ to STL (there are two choices), time to be at work (which may not be schedulable if the early flight is taken), time to be in transit in SJ (this is largely constrained by the choice of flight), and time to be at the theater in STL (which may be constrained if the later flight is taken). Value accrues in the amount of overlap of being at work during the workday, and also ceded in the amount of time being at the theater begins after 8pmCST.

The constraints are given conditionally on whether \( flight474 \) is scheduled, or \( flight213 \), and whether \( atWork \) is scheduled; this assumes all other schedulable intervals will be scheduled:

**general constraints:**

- \( 9amPST \) (s) workday
- \( STLtransit (< meets) atTheater \)

**if schedule:** \( atWork \)
- \( atWork (< meets) SJtransit \)

**if schedule:** \( flight474 \):
- \( SJtransit (< meets flight474 \)
- \( flight474 (< meets) STLtransit \)
- \( 1pmPST \) (s) \( flight474 \)
- \( 7:13pmCST \) (f) \( flight474 \)
- \( [1\text{hour}] (d) STLtransit \)
if schedule: \textit{flight213}:
    \textit{SJtransit} ($<$ meets) \textit{flight213}
    \textit{flight213} ($<$ meets) \textit{STLtransit}
    9amPST ($>$) \textit{flight213}
    3:30pmCST ($>$) \textit{flight213}
    [1hour] ($d$) \textit{STLtransit}
    [1hour] ($d$) \textit{SJtransit}

There are two value items:
    \( u(+) \) (tov tod tovi) (atWork, workday)
    \( u(-) \) (tbb) (8pmCST, atTheater)

In this form, it is possible to do a small amount of dominance reasoning. At the moment, there are still numerous distinguishable choices in history that the agent controls. Since \textit{atWork} is largely unconstrained in scenario class 1, its relation to \textit{workday} can be chosen: \( b, o, m, d \). As value is monotonically increasing in the overlap, the choices \( m \) and \( d \) are dominated and pruned.

**Comparing Value Items.**

In the choice between flights, of course, the positive value item must be traded against the negative value item. Qualitative ordering information does not provide much leverage on this tradeoff: we do not want merely to order goals lexicographically.
We fit linear functions to the times of the intervals involved in value items:

\[ u(+10)(\text{tov tod tovi})(A,B) \]

utility is increasing in the duration of \( A \) within \( B \) with a slope of 10: that is, if \( A \) overlaps \( B \), and the overlap is 0, contribution to utility is 0 utils; meanwhile, if overlap is 1 hour, contribution to utility is 10 utils.

We have just introduced quantities into the Allen representation. Have we deviated too far from the paradigm? The ontology still forces a conception of action, event, property, and now value, too, in terms of binary relations on intervals. What has changed is that vagueness regarding absolute times cannot be avoided everywhere. Anchoring intervals with respect to absolute time can still be avoided at the moment (we can say that \( STLtransit(>213)flight \) without saying when \( STLtransit \) actually begins, in CST or PST). But the ability to assess the tradeoff between the class 1 scenario and the class 2 scenario requires knowledge about the relative preference of three hours of work and thirteen minutes missed at the theater. It is traditionally assumed in DMUR that this tradeoff cannot reasonably be made by regarding it as a tradeoff between \( \text{atWork}(d b o)\text{workday} \) and \( 8\text{pmCST}(<)\text{atTheater} \). Subsequent comments regarding the likely form of probabilistic information further suggest that this introduction of quantities is unavoidable.

Given, then, that

\[ u(+1)(\text{tov tod tovi})(\text{atWork}, \text{workday}) \]
\[ u(-20)(\text{tbb})(8\text{pmCST}, \text{atTheater}) \]

the choice between missing thirteen minutes of the theater and spending three hours at work is clear. The preferred class of actions is class 2, since 0 is greater than \( (3)(1)-(13/60)(20) \); there is no basis for choosing between \( o \) and \( b \) as the relation between \( \text{atWork} \) and \( \text{workday} \).

We are anticipating the elicitation of these quantities to be similar to elicitation methods for multiattribute utility, assuming that the model adequately fits the agent’s preferences.

What we have done so far should be compared to what happens in DMUR with less complex temporal reasoning. Figure 4 shows a decision tree expressing the choices in our example. Note that all of the temporal reasoning (including some reasoning about the ordering of choice, which we discuss below) must be performed in advance of depicting the choice dependencies graphically.

**Uncertainty.**

DMUR also involves uncertainty measured by probability. The representation at this point is largely driven by the likely form in which probability information can be obtained. It matters importantly whether the probability distribution is reported as:

\[
\begin{align*}
\text{Prob} & (A(o)B) = A \\
& (A(b)B) = .5 \\
& (A(d)B) = .1
\end{align*}
\]
or as:

\[
\text{Prob} \begin{cases} 
[1\text{ hour]} (d) A & = .4 \\
[2\text{ hours}] (d) A & = .8 \\
[\infty\text{ hours}] (d) A & = 1; 
\end{cases}
\]

where the distinction is that the former is concerned with temporal relations and the latter is concerned with absolute times (the distinction is not that the former is marginal and the latter is cumulative). In the latter category, we include distributions over relations with absolute times, since these permit quantitative calculation.

We suspect that probabilistic information of the former kind is difficult to obtain. We provide for the latter kind of information. This is actually fortunate, since the former probabilistic information is useful with utility information of the form:

\[
\text{util} \begin{cases} 
A (o) B & = 12 \\
A (b) B & = 10 \\
A (d) B & = 8 
\end{cases}
\]

which is not the form implied by the use of value items with linear functions (if we used ramp functions, however, we could express such utilities).

Returning to the extended example, suppose that value does not refer to 8pmCST, but to BaryshnikovDances, which has an uncertain beginning time:

\[
\text{Prob} \begin{cases} 
8\text{pmCST (b) BDances} & = .5 \\
8:30\text{pmCST (b) BDances} & = .5 
\end{cases}
\]
Now in the calculation of value for any scenario of class 1 (where \textit{flight474} is scheduled), an expected utility can and must be taken. Bifurcation of the scenario is done on all (relevant, if this can be distinguished) events with measurable probabilities.

Maximizing the time at work and minimizing the transit times, the relevant comparison is 0 vs. (3hours)/(util/hour) – (.5)(13/60)(20) – (.5)(0)(20), which we have contrived to favor a class 1 scenario. Figure 5 shows this alteration. Note that we have bifurcated the scenario on the value item's condition (that \textit{BDances} \textless\textit{atTheater}) in order to evaluate whether the value item applies at all. Also, we have assumed that \textit{atTheater} actually gets scheduled, which is not necessarily the case (the alternative is to let value accrue positively in the (to\textit{v} to\textit{v} to\textit{v} to\textit{d}) of \textit{atTheater} and \textit{BDances}, which just looks more complicated).

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure5.png}
\caption{The extended example with an uncertainty about the time at which Baryshnikov begins dancing.}
\end{figure}

\textbf{Linear Programs.}

The linear form of value items permits the formulation of linear programming problems in certain situations. Basically, if it is known which guarding conditions of value items obtain, then
scheduling can be solved with a linear program. In order to fix the set of value items with satisfied guarding conditions, combinatorially numerous assumptions possibly need to be made. This is the approach of some modern operations research efforts, e.g., Rodin et al. [89], where a representationally expressive formalism is used, and a combinatorial search over classically solvable subproblems is performed (in Rodin’s case, a suite of 2-player differential games is created from a larger problem involving more than two players).

In our extended example, prior to introducing uncertainty, the l.p. is (we have not yet introduced absolute reference times nor dummy variables to eliminate inequalities):

\[
\text{class 1:} \\
\text{max} \ (1)(atWork_+ - 9amPST) - (20)(atTheater_- - 8pmCST) \\
s.t. \\
atTheater_- \leq STLtransit_+ \\
STLtransit_+ - STLtransit_- \geq 1 \\
STLtransit_- \geq flight474_+ \\
SJtransit_+ - SJtransit_- \geq 1 \\
\text{atWork}_+ \leq SJtransit_- \\
\text{flight474}_- = 1pmPST \\
\text{flight474}_+ = 7:13pmCST \\
\text{over} \\
atTheater_-, atTheater_+, STLtransit_-, STLtransit_+, SJtransit_-, SJtransit_+, atWork_-, atWork_+ \\
\text{class 2 (note the disappearance of the first term in the objective function):} \\
\text{max} \ - (20)(atTheater_- - 8pmCST) \\
s.t. \\
atTheater_- \leq STLtransit_+ \\
STLtransit_+ - STLtransit_- \geq 1 \\
STLtransit_- \geq flight213_+ \\
SJtransit_+ - SJtransit_- \geq 1 \\
\text{atWork}_+ \leq SJtransit_- \\
\text{flight213}_- = 9amPST \\
\text{flight213}_+ = 3:30pmCST \\
\text{over} \\
atTheater_-, atTheater_+, STLtransit_-, STLtransit_+, SJtransit_-, SJtransit_+, atWork_-, atWork_+ \\
\]

Once the event is introduced (BaryshnikovDances), there are twice as many linear programs to solve, and events in general cause combinatorial expansion.

**Metaphor: Clothespins on Timelines.**

One of the attractions of decision theory, especially after Raiffa [68], was the indelible image of a decision tree. The mental picture of a decision tree helped structure one’s thinking about decisions. A tree annotated with probabilities and utilities was simple to draw; it was an effective
tool for communication whether it implied a useful computation (which it did: backward induction or dynamic programming) or not.

More complex temporal reasoning in planning destroys the image. Allen would have us imagine an unstructured soup of time intervals and a constraint network imposed by the agent's control. The usual conceptions of decision tree (with increasing time, as in game trees or discrete sequential control) becomes a restricted special case, with its sequence of discrete time that was usually the most convenient sequence for conditioning actions and events. The tree in figure 4 is inadequate because it obscures all of the interesting problem structure. Similarly, the economist's utility contours in Cartesian space for 2-commodity utility are useful only when the problem of sequencing actions to achieve a particular outcome in the space is uninteresting. A decision theorist concerned with uncertainty would object to picturing merely the utility space, since all of the complexity of the problem is in how to get to locations in the space. We would like to find a useful mental image that replaces the decision tree.

Something along the lines of figures 3 and 5 would seem to be needed, with some embellishment.

We ask the framer of this kind of decision to picture a series of clotheslines on which clothespins are to be placed. There is a separate line for each property whose occurrence is interesting. In our problem, there is a separate line for each time interval. In a problem in which a property is intermittent, such as scheduling the entry and exit of Larry Bird from a basketball game, there will be a single line for \textit{LarryBirdIn}, on which several pairs of clothespins will be placed. When a clothespin's location is not determined with certainty, we imagine that a blanket is covering it (this presumes a certain contiguity of the possibilities unless blankets can have holes; also, some blankets are quite long). The clothes that are pinned by the clothespins constrain the relative locations of pins on different lines (the shapes are quite odd, since lines can interfere arbitrarily, even though the physical model requires locality of interactions). Various shadows are cast over the series of lines and the configuration of clothes; some overlaps are good, others are bad, and goodness may depend on the amount of exposure to the sun. Imagine a time in suburban America when all households have essentially the same laundry to hang in essentially the same sunning conditions. The problem is first to consider all of the possible ways that the different houses in the neighborhood can arrange their clothespins. Then the problem is to determine to which household best to belong.

Discussion.

The main problem with this approach has to do with the combinatorics of conditioning.

As indicated, value items have guarding conditions; the combinations in which they can be satisfied cause a proliferation of cases. We did not discuss conditional probabilities. In another example, of scheduling Larry Bird's entry to and exit from a basketball game, the interactions of the probabilities at which \textit{4thFoul} and \textit{LarryBirdTired} occur are non-trivial (the probability that he grows tired in the next two minutes depends on whether he has received his fourth foul, and the probability of his receiving his fourth foul in the next two minutes depends on whether he is tired); we begin to have the problem of describing general stochastic processes with too simple a
language. Conditioning probability information on the relations between time intervals requires more proliferation of cases.

Moreover, there is a hidden combinatorial cost in developing scenarios, since the best order in which to condition action (or the best order in which to impose constraints) is implicit in the problem specification (unlike in a decision tree, the real purpose of which is to make this order explicit). In our example, we identified the most important choice as a choice between flight213 and flight474, and we cheated: we should instead have said that there were an inSJ and inSTL properties and that the exclusion of the two flights was implied by constraints on location. If we do not know in advance what sequence or sequences of choices permit all interesting cases to be considered, then we have to start making scheduling choices blindly, look at all possible subsequent choices given that choice, and backtrack to consider other possible starting choices. For example, beginning our example scheduling problem with the various atWork possibilities, the flight213 possibilities are never explored. Beginning with the choice between flights, the numerous atWork possibilities are finessed because the value item can be maximized; there is no local extremum reached if that is the last interval to be fixed. This is an order-dependence that we appear to have introduced only because the introduction of value allows whole classes of possibilities to be pruned via dominance, and the determination of dominance appears to be sensitive to order.

Another subtlety is the prospect of default utility. Scenarios are usually classes of scenarios where not all extents have been fixed. Again, this natural vagueness is a feature of the time interval representation. However, value items apply to, for instance, overlapping intervals when the intervals stand in the overlapping relation. When the intervals do not stand in that relation, there is an ambiguity: are they constrained not to stand in that relation, or has the scenario not been specified to the level of detail that would exhibit that relation? Officially in the DMUR conception of utility, utility can be computed only for scenarios that have been fully specified in every value-relevant way (otherwise, expectations are supposed). This is a problem for systems performing temporal reasoning at this high level of abstraction. Loui [88,89,90] says that this kind of abstraction is one way to evaluate utility heuristically, but there seem to be pitfalls: value items farther in the future may be more important, but hard to fix in a scenario; meanwhile, abstraction of this kind supposes that the more important value items, all things being equal, will be processed first.

The clothespin-dropping metaphor is useful in practice. Our initial implementation maintains temporal relations by dropping clothespins on timelines that are scaled by reference times. In building a scenario, intervals are anchored at reference times (e.g., 8pm CST). The expression of constraint (preconditions of actions or events) and value remains vague, in terms of Allen’s thirteen relations on intervals. But actual scheduling of actions and events refers to reference times. This is possible because we require that scheduling actions refer to reference times, and events’ probability distributions are also precise about absolute time. This vagueness is encouraged in problem specification, not in computation of the problem’s solution.

Finally, we must consider whether this DMUR extension to temporal interval representation remains in paradigm. This depends on what was considered to be attractive about Allen’s framework in the first place. We have introduced quantities and made conditioning much more impor-
tant. The constraint–propagation that dominated Allen’s understanding of planning in this representation is the most trivial part of the reasoning we must now perform. But the emphasis on relations between intervals, the ontology, remains. Intervals’ endpoints are more important in our extension, because of the quantization. But we still ask the problem to be framed in terms of intervals, which permits expression of complex temporal relations such as non-linearity and simultaneity of action.
Bibliography.