Satisficing Evaluation Functions: The Heart of the New Design-to-Criteria Paradigm *

Thomas Wagner
Computer Science Department
University of Massachusetts
Amherst, MA 01003
Email: wagner@cs.umass.edu

Alan Garvey
Computer Science Department
Pacific Lutheran University
Tacoma, WA 98447
Email: garveyaj@plu.edu

Victor Lesser
Computer Science Department
University of Massachusetts
Amherst, MA 01003
Email: lesser@cs.umass.edu

UMass Computer Science Technical Report 1996-82

November 16, 1996

Abstract

Design-to-Criteria scheduling is the process of custom tailoring a way to achieve a high-level task, via actions described in a TÆMS model of the task, to fit a particular client's quality, cost, and duration criteria or needs. At the heart of the Design-to-Criteria paradigm is the ability to determine how well a particular schedule, or schedule abstraction, fits a set of design criteria. The process of measuring "goodness" of schedules or alternatives and determining which items are best is called *evaluation*. Working to meet criteria is ubiquitous in the Design-to-Criteria scheduling system and consequently evaluation is used at every turn. The new evaluation functions operate to determine a principled measurement of goodness based on *relativity* and *proportionality*. Relativity is important because the objective is to make satisficing choices and the goodness of one option is relative to the other possible options. Proportionality is a major concern because we do not want different quality, cost, and duration scales to skew the evaluation mechanism and because the client's criteria is described in a relative/proportionalistic fashion. The new evaluation functions are paired with a new criteria specification metaphor, *importance sliders*. The slider metaphor enables clients, users or other systems, to define the relative importance of quality, cost, and duration with respect to three classes of concerns: raw goodness, thresholds and limits, and uncertainty.

*This material is based upon work supported by the National Science Foundation under Grant No. IRI-9523419, the Department of the Navy, Office of the Chief of Naval Research, under Grant No. N00014-95-1-1198, and via a subcontract from Boeing Helicopter which is being supported by DARPA under the RaDEO program (contract number 70NANB6H0074). The content of the information does not necessarily reflect the position or the policy of the Government, National Science Foundation, or Boeing Helicopter and no official endorsement should be inferred.

1 Introduction

TÆMS (Task Analysis, Environment Modeling, and Simulation) [1, 2, 3] is a task modeling framework used to describe and reason about complex problem solving processes. The model explicitly represents different alternative approaches to achieving a task, how they interact, and the quality, cost, and duration trade-offs of different actions. Design-to-Criteria [8] scheduling is the process of custom tailoring a way to achieve the high-level task, via the actions described in the model, to fit a particular client's quality, cost, and duration criteria or needs. At the heart of the Design-to-Criteria paradigm is the ability to determine how well a particular schedule, or schedule abstraction called an *alternative*, fits a set of design criteria. The process of measuring "goodness" of schedules or alternatives and determining which items are best is called *evaluation*. Working to meet criteria is ubiquitous in the Design-to-Criteria scheduling system. Evaluation is used to prune alternatives from alternative sets when the sets grow too large and the scheduling problem becomes intractable. It is also used to determine what alternatives to turn into schedules and to decide which completed schedule best satisfices to meet the criteria.

The main predecessor to Design-to-Criteria, Design-to-Time [5, 6, 7], focuses on quality/time trade-offs in a heuristic fashion. While heuristics still play a role in Design-to-Criteria, the design specifications or evaluation criteria that describe the desired quality, cost, and duration trade-offs are used at every major decision point. In Design-to-Time, the design to specification component is an add-on at the back-end of the scheduling process – it is only used to select the "best" schedule and quality/time-centric heuristics govern the entire alternative selection process without taking into consideration the design/evaluation criteria. Furthermore, previous evaluation functions suffered from lack of proportionality, inconsistency, and required acrobatics on the part of the client to obtain the desired results. The new evaluation functions operate to determine a principled measurement of goodness based on *relativity* and *proportionality*. Relativity is important because the objective is to make satisficing choices and the goodness of one option is relative to the other possible options. Proportionality is a major concern because we do not want different quality, cost, and duration scales to skew the evaluation mechanism and because the client's criteria is described in a relative/proportionalistic fashion. The new evaluation functions are paired with a new criteria specification metaphor, *importance sliders*. The slider metaphor enables clients, users or other systems, to define the relative importance of quality, cost, and duration with respect to three classes of concerns: raw goodness, thresholds and limits, and uncertainty.

In this document we define the new evaluation functions and and the new criteria specification metaphor. Readers unfamiliar with TÆMS or Design-to-Time / Design-to-Criteria scheduling should consult more foundational work as the intent here is to focus on the crux of the paradigm – the evaluation process.

2 Sliders - The Client Criteria Specification Metaphor

The objective of the evaluation functions is to translate a client's needs, expressed as evaluation criteria, into choosing the course of action that best meets the criteria. We subscribe to the notion that clients are good at expressing and reasoning about the *relative* importance of quality, cost, and duration, but that they are less good at assigning particular absolute values that denote goodness. Thus, our evaluation functions operate on the conceptual notion of *importance sliders* that clients "set" for each dimension in the criteria set. The importance sliders, which take on values over a particular small integer range, say zero to ten, describe the relative importance of each of dimension in a domain independent fashion. Using the sliders, client applications or users can express the

notions like "quality is twice as important as cost and duration is half as important," or "quality and duration are equally important but cost is no issue."

While we have introduced sliders in a general sense there are actually four sets of sliders used in the criteria specification process, some of which are accompanied by absolute requirements in the form of thresholds or limits. The slider sets, shown in Figure 1, are:

Raw Goodness This slider set contains sliders for each dimension, quality, cost, and duration. Its purpose is to describe the relative importance of each dimension. For example, setting quality to "10" and cost and duration to "5" expresses the notion that quality is twice as important as each of the other dimensions and that it should weigh twice as heavily as each when evaluating schedules or alternatives. Alternately, this says that cost and duration combined are equally as important as quality.

Threshold and Limits This slider set also contains sliders for each dimension, however, in this set each slider is paired with an absolute value that describes thresholds or limits for the particular dimension. This set allows clients to set minimum desired quality thresholds and maximum duration and cost limits and then to describe how important these limits are relative to each other. Note we do not mean threshold or limit in the hard constraint sense. Limits and thresholds describe quantities that schedules or alternatives must beat in order to get points from this set of sliders, i.e., schedules that fail to beat thresholds and limits may still be returned for execution. We use the term *hard constraint* to denote limits and thresholds that must be meet in order for a schedule to be considered for execution. We address the issue of satisficing with respect to hard constraints in Section 6.

At first glance, it is intuitive to think that the sliders in this case should be omitted and that the relative importances described in the raw goodness slider set should be used instead. While the relative importances expressed in the two sets of sliders may be identical, this separation allows the client to specify concepts like "Cost, quality and duration are equally important in general, but schedules whose quality is over my threshold are particularly important."

The first aspect of this example is expressed by setting all sliders of the raw goodness set to the same value. The second aspect of this example, that schedules whose quality is above a certain threshold are preferred, is expressed by setting the quality threshold to the desired value and then setting its slider to a high value relative to the cost and duration threshold/limit sliders.

Uncertainty Whereas the other slider sets address quality, cost, and duration issues, this slider set describes how important uncertainty is to the client. In particular applications it may be more desirable to pick a slower, more costly schedule that returns lower expected quality because the certainty about these values is very high. This slider set thus contains a slider for each dimension, quality, cost, and duration and it defines how important reducing uncertainty in each dimension is relative to the other dimensions.

Meta This slider set relates the importance of the three previous slider sets. This allows clients to focus on relating quality, cost and duration with each other in each of the cases above then to "step back" and decide how important each of the different aspects are relative to each other. In this slider set, sliders take on percentage values from 0 to 100 with the constraint that the sum of the sliders must be equal to 100%.

In the example slider set, shown in Figure 1, quality is the most important general factor with cost being one half as important and duration being not important at all. In terms of thresholds, quality and duration have none, but schedules whose cost is below \$5.75 are preferred. Schedules

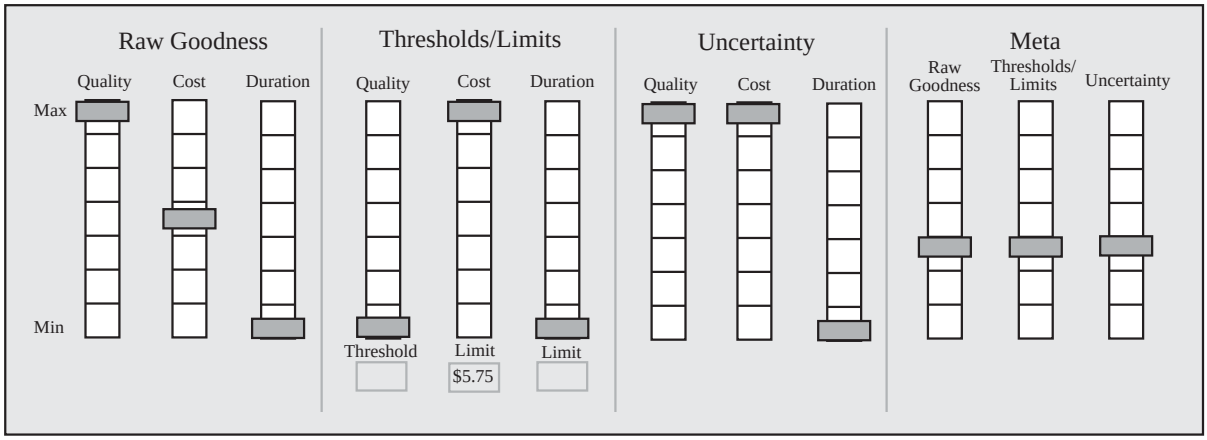

Figure 1: A Slider Set Describing Particular Criteria

whose expected quality and cost values are more certain are also preferred and uncertainty about duration is not an issue. Relating the three sets of criteria together, they are all equally (33%) important and thus all contribute equally to the overall ranking. Mapping this example to the real world, this could describe the criteria of an individual performing research on the web who does not need the information in a timely fashion, has only seven dollars in his or her pocket, wants good quality information, but also wants to be fairly certain of the cost and quality of the proposed solution before committing to a course of action.

3 Mapping Sliders to Ratings

After defining the slider sets, the problem then becomes how to relate them in such a way that the evaluation function results match expectations. When determining schedule or alternative "goodness," alternatives or schedules are rated using the relative importances expressed on the sliders. We associate a rating component with each of the slider banks, excluding the meta bank, and then combine them according to the relative weights expressed in the meta slider bank. The omnipresent themes in the rating calculations are relativity and proportionality.

In general, we calculate the rating component for a given slider bank by calculating sub-components for each dimension, quality, cost, and duration. Each dimension's sub-component is computed by looping over the set of items to be evaluated and normalizing each item's expected value or expected probability (in the uncertain case) for that particular dimension, and then multiplying the result by the relative importance as expressed in the slider. It is crucial to normalize the values to a common scale so that in domains where one dimension, say quality, is exponentially larger than the others, cost and duration, it does not dominate the ratings disproportionately. Normalization based on the observed minimum and maximum values for a given dimension is similarly important. We are at all times interested in relative goodness between alternatives or schedules. By using minima and maxima that are derived from the set of items being rated, we automatically scale the grain size to define relative differences in the items. For example, say Schedule A has expected quality of 4.7, Schedule B has expected quality of 4.2, and Schedule C has expected quality of 4.0. In absolute numerical terms Schedule A is "a little" better than both B and C. However, in relative terms, Schedule A is by far the best of the possible schedules. This notion of relative scaling will become more clear from the equations that follow.

We calculate the rating component for the first slider bank, that describes the raw goodness of

a particular dimension, as follows:

1. Get min and max expected values for quality, cost, and duration that occur in the set of schedules or alternatives being rated, i.e., look through the entire set and note the min and max expected values for each dimension.

2. Loop over the set of alternatives or schedules to be rated and calculate the raw goodness rating for each by calculating the quality, cost, and duration sub-components as follows in Steps 3 through 5.

3. Let *this* denote the alternative or schedule under consideration. Its quality sub-component is a function of the raw goodness quality slider value and the percentage of quality achieved by *this* relative to the best quality for all alternatives or schedules in the set.

$$rating_{quality} = \frac{(this.expected\_quality - quality\_min)}{quality\_max - quality\_min} * \frac{quality\_slider\_value}{total\_points\_in\_raw\_goodness\_bank}$$

4. Duration is different than quality in that more duration is a less good thing. Whereas with the quality related equation achieving the best quality of all items in the set should bring the highest reward, in this case, achieving the lest duration of all items in the set should bring the highest reward. Obtaining the least reward is likewise reversed. Alternatives or schedules that take more time should get lower rewards.

$$rating_{duration} = \frac{(duration\_max - this.expected\_duration)}{duration\_max - duration\_min} * \frac{duration\_slider\_value}{total\_points\_in\_raw\_goodness\_bank}$$

5. Cost is like duration in that lower cost is better. Thus the equation is similarly reversed.

$$rating_{cost} = \frac{(cost\_max - this.expected\_cost)}{cost\_max - cost\_min} * \frac{cost\_slider\_value}{total\_points\_in\_raw\_goodness\_bank}$$

6. The quality, duration, and cost sub-components are then summed to obtain the aggregate raw goodness rating component.

The threshold or limit rating component is likewise composed of three sub-components. Originally, we modified the equations above by replacing the derived quality minimum, and the derived cost and duration maximums, with the client provided threshold/limits. However, this approach leads to rewards for high relative quality, and low relative cost and duration, from both the threshold/limit bank and from the raw goodness bank. The resulting ratings often didn't map well to the semantic model presented by the sliders. Thus, the current threshold/limit rating components are even more simple to compute – quality at or above the specified threshold, and cost and duration at or below the specified limits, are rewarded according to the relative settings of the quality, cost, and duration sliders. Beating a threshold or a limit is rewarded the same regardless of how well a particular schedule or alternative beats the threshold or limit. The threshold rating component is computed as follows:

1. For each member of the set of schedules or alternatives to evaluate, calculate the quality, duration, and cost rating sub-components using the threshold/limit sliders and their associated client supplied thresholds or limits as described in Steps 2 through 4.

2. To calculate the quality component, we ascertain whether or not the expected quality of the item being rated is above the specified threshold. If so, the rating component gets the points alloted to the quality slider of the threshold/limit bank. Members that do not achieve the minimum threshold get no reward.

$$
\begin{aligned}
&if\ \ (this.expected\_quality \geq client\_quality\_threshold)\ \ then \\
&\qquad rating_{quality} = \frac{quality\_slider\_value}{total\_points\_in\_threshold\_bank} \\
&else \\
&\qquad rating_{quality} = 0
\end{aligned}
$$

3. Unlike the quality calculation in which the client supplies the minimum threshold, in the duration calculation the client supplies the maximum limit. Set members whose duration is above the client supplied limit receive no rating and those whose duration is under the limit receive the proportion of points allocated to the duration limit slider.

$$
\begin{aligned}
&if\ \ (this.expected\_duration \leq client\_duration\_limit)\ \ then \\
&\qquad rating_{duration} = \frac{duration\_slider\_value}{total\_points\_in\_threshold\_bank} \\
&else \\
&\qquad rating_{duration} = 0
\end{aligned}
$$

4. As with duration, the client supplies a desired cost limit and set members that exceed the limit receive no points.

$$
\begin{aligned}
&if\ \ (this.expected\_cost \leq client\_cost\_limit)\ \ then \\
&\qquad rating_{cost} = \frac{cost\_slider\_value}{total\_points\_in\_threshold\_bank} \\
&else \\
&\qquad rating_{cost} = 0
\end{aligned}
$$

5. The threshold/limit rating component is a sum of the $rating_{quality}$, $rating_{duration}$, and $rating_{cost}$ sub-components.

The uncertainty rating component is different from the previous components because it does not look at quality, cost, and duration values, but at the uncertainty associated with these values. The uncertainty component, while somewhat more daunting conceptually, is actually straightforward to compute and to understand. Consider the quality case. The general idea is to reward alternatives or schedules based on how likely it is that their expected quality value, or one better, will actually occur.[1] Thus we compute the probability that the quality, as expressed by the discrete probability distribution, is either greater than or equal to the expected value, we then normalize and scale the probability as with the previous components, and finally multiply by the proportion of points

[1] An alternate approach to interpreting the issue of reduced uncertainty is to determine the probability that the actual value will fall near the expected value, on the upside or the downside, and reward accordingly. Our somewhat less intuitive view of determining the conservativeness of the expected value is partly driven by the interpretation of uncertainty reduction as denoting a desire not to get results worse than advertised. This is also related to a desire not to reschedule. As the scheduler often performs calculations with expected values, durations and costs under the expected values, and quality above the expected value, will only improve the schedule and will not result in rescheduling due to time or cost overrun or a quality underrun. If the definition of uncertainty reduction should evolve or change, the equations need not change greatly. Only the probability factor used in the equations, i.e., its calculation, must change to reflect any new interpretations.

allocated to the uncertainty quality slider. Consider a partial example, if an alternative has a simple quality distribution that denotes 25% of the time 0 quality will result and 75% of the time quality 10 will result, its resulting expected quality value is 7.5. Contrast this with an alternative whose quality distribution denotes that 50% of the time 0 quality will result and 50% of the time 15 quality will result; its expected quality is also 7.5. However, the probability that the first alternative will generate a quality value greater than or equal to the expected value is .75 whereas the second alternative's probability is only .50. This is the gist of the uncertainty rating sub-components – the more certain that the expected value or one better will occur, the more reward. The specific rating system follows:

1. Find the min and max probabilities for quality, cost, and duration by examining all the members in the set of alternatives or schedules to be rated. Note, this entails calculating and recording the probability that the quality is greater than or equal to the expected value, cost is less than or equal to the expected value, and duration is less than or equal to the expected value for *each* member of the set. Then marching through the set and finding the min and max values of the probabilities for each dimension.

2. For each item in the set of alternatives or schedules being evaluated, calculate the rating sub-components as described in Steps 3 through 5.

3. Calculate the quality sub-component by first finding the probability that the quality value is greater than or equal to the expected value and then multiplying by the uncertainty quality slider value scaled by the number of uncertainty points in play.

$$rating_{quality} = \frac{(Prob(this.quality\_value >= this.expected\_quality\_value) - quality\_probability\_min)}{quality\_probability\_max - quality\_probability\_min} * \frac{quality\_slider\_value}{total\_points\_in\_uncertainty\_bank}$$

4. The duration sub-component is calculated in a similar fashion. However, the notion that less duration is better translates into using the probability that the duration value is actually less than or equal to the expected value, rather than greater than or equal to, i.e., better to have the expected value be an overestimate.

$$rating_{duration} = \frac{(Prob(this.duration\_value <= this.expected\_value) - duration\_probability\_min)}{duration\_probability\_max - duration\_probability\_min} * \frac{duration\_slider\_value}{total\_points\_in\_uncertainty\_bank}$$

5. The cost sub-component is similarly computed.

$$rating_{cost} = \frac{(Prob(this.cost\_value <= this.expected\_value) - cost\_probability\_min)}{cost\_probability\_max - cost\_probability\_min} * \frac{cost\_slider\_value}{total\_points\_in\_uncertainty\_bank}$$

6. The uncertainty rating component is a sum of the $rating_{quality}$, $rating_{duration}$, and $rating_{cost}$ sub-components.

After computing the raw goodness, threshold/limit, and uncertainty rating components, the alternate or schedule rating is computed by weighting the rating components according to the relations specified by the meta sliders. For example, if the raw goodness slider set is given full

weight, the other rating components will contribute zero to the overall rating. The general equation is simple:

$$\begin{aligned} overall\_rating &= raw\_goodness\_rating * meta\_raw\_goodness\_slider\_value \\ &+ threshold\&limit\_rating * meta\_threshold\&limit\_slider\_value \\ &+ uncertainty\_rating * meta\_uncertainty\_slider\_value \end{aligned}$$

4 Putting it All Together - An Example

Let us revisit the sample slider set presented in Section 2 and demonstrate its application to the evaluation and rating of four schedules. We will not explore the internals of the schedules, but only describe them in terms of their aggregate quality, cost, and duration distributions – the same grain size used by the schedule evaluation functions.

- Schedule A has distributions as follows:

  Quality: (10% 0)(10% 5)(30% 7)(50% 20), $E(\overline{X}) = 12.6$
  Cost: (20% \$3)(20% \$4)(20% \$5)(20% \$6)(20% \$7), $E(\overline{X}) = 5$
  Duration: (50% 10 minutes)(50% 20 minutes), $E(\overline{X}) = 15$

- Schedule B has distributions:

  Quality: (10% 0)(90% 10), $E(\overline{X}) = 9$
  Cost: (40% \$3)(40% \$4)(20% \$16), $E(\overline{X}) = 6$
  Duration: (50% 4 minutes)(50% 8 minutes), $E(\overline{X}) = 6$

- Schedule C has distributions:

  Quality: (10% 1)(90% 9), $E(\overline{X}) = 8.2$
  Cost: (50% \$4)(50% \$5), $E(\overline{X}) = 4.5$
  Duration: (50% 7 minutes)(50% 8 minutes), $E(\overline{X}) = 7.5$

- Schedule D has distributions:

  Quality: (10% 9)(40% 10)(50% 12), $E(\overline{X}) = 10.9$
  Cost: (65% \$4)(10% \$5)(25% 6), $E(\overline{X}) = 4.6$
  Duration: (40% 12 minutes)(40% 13 minutes)(20% 15 minutes), $E(\overline{X}) = 13$

The schedules above have some very definite trade-offs. Schedule A has a wider range of possible qualities but its highest possible quality is around twice the maximum quality possible from the other schedules. Schedule A's costs range from \$3 to \$7 and all values are equally probable. Schedule B's costs are expected to fall near the same value, but there is a possibility that the cost will be much greater than Schedule A's cost. Schedule C's expected cost is similar to A's and B's, but it lacks the high-end potential as does Schedule D to a lesser extent. Duration-wise, Schedule A requires significantly more time to execute than either Schedule B or Schedule C, and slightly more than Schedule D.

We begin by finding the min and max expected values for all dimensions of the three schedules. They are: quality min 8.2, quality max 12.6, cost min 4.5, cost max 6, duration min 6, and duration max 15. Using these values, we compute the raw goodness rating components via the quality, cost, and duration subcomponents:

Schedule ?: $rating_{goodness} = rating_{quality} + rating_{cost} + rating_{duration}$

Schedule A:

$rating_{goodness\_quality\_subcomponent} = \frac{12.6-8.2}{12.6-8.2} * \frac{slider\_value}{total\_points\_in\_bank} = 1 * \frac{10}{15} = .66$

$rating_{goodness\_cost\_subcomponent} = \frac{6-5}{6-4.5} * \frac{slider\_value}{total\_points\_in\_bank} = .67 * \frac{5}{15} = .22$

$rating_{goodness\_duration\_subcomponent} = \frac{15-15}{15-6} * \frac{slider\_value}{total\_points\_in\_bank} = 0 * \frac{0}{15} = 0$

Thus, $rating_{goodness\_component} = .88$

Schedule B:

$rating_{goodness\_quality\_subcomponent} = \frac{9-8.2}{12.6-8.2} * \frac{slider\_value}{total\_points\_in\_bank} = .18 * \frac{10}{15} = .12$

$rating_{goodness\_cost\_subcomponent} = \frac{6-6}{6-4.5} * \frac{slider\_value}{total\_points\_in\_bank} = 0 * \frac{5}{15} = 0$

$rating_{goodness\_duration\_subcomponent} = \frac{15-6}{15-6} * \frac{slider\_value}{total\_points\_in\_bank} = 1 * \frac{0}{15} = 0$

Thus, $rating_{goodness\_component} = .12$

Schedule C:

$rating_{goodness\_quality\_subcomponent} = \frac{8.2-8.2}{12.6-8.2} * \frac{slider\_value}{total\_points\_in\_bank} = 0 * \frac{10}{15} = .0$

$rating_{goodness\_cost\_subcomponent} = \frac{6-4.5}{6-4.5} * \frac{slider\_value}{total\_points\_in\_bank} = 1 * \frac{5}{15} = .33$

$rating_{goodness\_duration\_subcomponent} = \frac{15-7.5}{15-6} * \frac{slider\_value}{total\_points\_in\_bank} = .83 * \frac{0}{15} = 0$

Thus, $rating_{goodness\_component} = .33$

Schedule D:

$rating_{goodness\_quality\_subcomponent} = \frac{10.9-8.2}{12.6-8.2} * \frac{slider\_value}{total\_points\_in\_bank} = .61 * \frac{10}{15} = .41$

$rating_{goodness\_cost\_subcomponent} = \frac{6-4.6}{6-4.5} * \frac{slider\_value}{total\_points\_in\_bank} = .93 * \frac{5}{15} = .31$

$rating_{goodness\_duration\_subcomponent} = \frac{15-13}{15-6} * \frac{slider\_value}{total\_points\_in\_bank} = .22 * \frac{0}{15} = 0$

Thus, $rating_{goodness\_component} = .72$

At this point we can see that Schedule A is the best schedule in terms of the raw goodness preferences expressed by the criteria because it achieves the observed quality maximum while holding cost to the lower end of the cost spectrum. Schedule D also does well since its quality is close to the maximum while its cost is very near the minimum. If duration were a small issue, expressed by a small percentage allocated to the duration raw slider, Schedule D would surpass schedule A because of its lower expected duration in conjunction with high quality and low cost. In contrast to Schedules A and D, that achieve high quality, Schedule C achieves the lowest quality and thus receives no points for raw quality. However, Schedule C also has the lowest cost and thus obtains all the points allocated to the cost slider. Schedule C also has the second lowest duration, but since raw duration is not important to the client, its duration goodness goes unrewarded. Schedule B is the end runner for its low quality and high cost and because duration is not a factor.

Notice that if the bounds on the quality dimension shifted, i.e., Schedule A's quality was a little lower or Schedule C's quality a little lower, Schedule D could equal or surpass Schedule A in the raw-goodness category. However, even if Schedule D surpassed Schedule A, they would both would continue to dominate Schedule C in this category, which in turn would continue to dominate Schedule B. Now consider the threshold/limit rating sub-components:

Schedule A:

$5 \leq 5.75$, thus $rating_{limit\_cost\_subcomponent} = \frac{slider\_value}{total\_points\_in\_bank} = \frac{10}{10} = 1$

Thus, $rating_{limit\_component} = 1$

Schedule B:

$6 \not\leq 5.75$, thus $rating_{limit\_cost\_subcomponent} = 0$

Thus, $rating_{limit\_component} = 0$

Schedule C:

$4.5 \leq 5.75$, thus $rating_{limit\_cost\_subcomponent} = \frac{slider\_value}{total\_points\_in\_bank} = \frac{10}{10} = 1$

Thus, $rating_{limit\_component} = 1$

Schedule D:

$4.6 \leq 5.75$, thus $rating_{limit\_cost\_subcomponent} = \frac{slider\_value}{total\_points\_in\_bank} = \frac{10}{10} = 1$

Thus, $rating_{limit\_component} = 1$

All schedules whose cost is below the specified limit of \$5.75, namely A, C, and D, receive 100% of the points allocated to the cost limit/threshold slider. Schedule B's cost exceeds the limit and B accordingly receives no reward from the limit/threshold slider bank.

Finally we consider the uncertainty factors. In order to do this, we must first calculate the probability for each schedule, each dimension, that the value returned will be better than or equal to the expected value. For Schedule A, the probabilities are: quality .5, cost .6, and duration .5. For Schedule B: quality .9, cost .8, and duration .5. Schedule C: quality .9, cost .5, and duration .5. Schedule D: quality .5, cost .65, duration .8. Then we determine the minima and maxima for each dimension. They are: quality min .5, quality max .9, cost min .5, cost max .8, duration min .5, and duration max .8.

Schedule A:

$rating_{uncertainty\_quality\_subcomponent} = \frac{.5-.5}{.9-.5} * \frac{slider\_value}{total\_points\_in\_bank} = 0 * \frac{10}{20} = 0$

$rating_{uncertainty\_cost\_subcomponent} = \frac{.6-.5}{.8-.5} * \frac{slider\_value}{total\_points\_in\_bank} = .33 * \frac{10}{20} = .17$

Thus, $rating_{uncertainty\_component} = .17$

Schedule B:

$rating_{uncertainty\_quality\_subcomponent} = \frac{.9-.5}{.9-.5} * \frac{slider\_value}{total\_points\_in\_bank} = 1 * \frac{10}{20} = .5$

$rating_{uncertainty\_cost\_subcomponent} = \frac{.8-.5}{.8-.5} * \frac{slider\_value}{total\_points\_in\_bank} = 1 * \frac{10}{20} = .5$

Thus, $rating_{uncertainty\_component} = 1$

Schedule C:

$rating_{uncertainty\_quality\_subcomponent} = \frac{.9-.5}{.9-.5} * \frac{slider\_value}{total\_points\_in\_bank} = 1 * \frac{10}{20} = .5$

$rating_{uncertainty\_cost\_subcomponent} = \frac{.5-.5}{.8-.5} * \frac{slider\_value}{total\_points\_in\_bank} = 0 * \frac{10}{20} = 0$

Thus, $rating_{uncertainty\_component} = .5$

Schedule D:

$rating_{uncertainty\_quality\_subcomponent} = \frac{.5-.5}{.9-.5} * \frac{slider\_value}{total\_points\_in\_bank} = 0 * \frac{10}{20} = 0$

$rating_{uncertainty\_cost\_subcomponent} = \frac{.65-.5}{.8-.5} * \frac{slider\_value}{total\_points\_in\_bank} = .5 * \frac{10}{20} = .25$

Thus, $rating_{uncertainty\_component} = .25$

For this round of ratings Schedule B obtains all the possible points because it has the least uncertainty, relative to the other schedules, about the likelihood of getting a value equal to, or better than, its expected value in the quality and cost dimensions. Astute readers will notice that Schedule C failed to get any points for uncertainty in cost – this is counterintuitive if the cost distributions for the schedules are consulted. Schedule C actually has the narrowest range of possible costs. However, as discussed previously, the issue with uncertainty is the probability that the value returned by the schedule will be as good as or better than the expected value. In C's case, 50% of the time the expected cost of \$4.5 or lower will result and 50% of the time the cost will be higher. With Schedule B, 80% of the time its cost will be lower than or equal to the expected cost. Thus B obtains more points. Actually, since B has the least relative uncertainty with respect to cost or quality it obtains all the points in both cases. Schedule D is mixed with respect to uncertainty; 50% of the time its quality will be less than expected and 35% of the time

its cost will be greater than expected. Schedule A is clearly the looser as 50% of the time its quality will be less than expected and 50% of the time its cost will be higher. Schedule A is not on par with the other schedules in this respect and consequently gets very few points for certainty.

While micro examination is useful to understand the application of the component equations, consider the aggregate ratings as calculated using the weights from the meta bank of sliders. In this particular case all three components, raw goodness, thresholds/limits, and uncertainty, contribute one third to the final rating.

Schedule A: $rating = .88 * 33\% + 1 * 33\% + .17 * 33\% = .68$
Schedule B: $rating = .12 * 33\% + 0 * 33\% + 1 * 33\% = .37$
Schedule C: $rating = .33 * 33\% + 1 * 33\% + .5 * 33\% = .60$
Schedule D: $rating = .72 * 33\% + 1 * 33\% + .25 * 33\% = .65$

Overall, Schedule A best meets the criteria. Since each component contributes one third to the total points, A's dominance in raw-goodness and its 100% in the limit/threshold category more than compensates for its less than impressive certainty rating. Schedules C and D each just miss the prize (relative to B) but for different reasons. Schedule D has a poor showing in the uncertainty category relative to C and B. Schedule C has a poor showing in the raw goodness category relative to D and a moderate showing in the uncertainty category. Schedule B, which is a strong contender in uncertainty alone comes in a weak fourth.

Let us consider the same schedules with a slightly different slider setting. In this case we will move the meta slider for uncertainty to 50% and give each of the threshold/limit and raw-goodness components 25% of the total weight. We do not have to recompute any of the component ratings, only the final weights.

Schedule A: $rating = .88 * 25\% + 1 * 25\% + .17 * 50\% = .56$
Schedule B: $rating = .12 * 25\% + 0 * 25\% + 1 * 50\% = .53$
Schedule C: $rating = .33 * 25\% + 1 * 25\% + .5 * 50\% = .58$
Schedule D: $rating = .72 * 25\% + 1 * 25\% + .25 * 50\% = .64$

In this scenario, where reducing uncertainty is paramount and raw goodness and beating the cost threshold are lesser concerns, Schedule D dominates because of its reasonable showing in certainty and strong showing in both raw-goodness and limits/thresholds. Schedule C brings in a respectable second because of its strong showing in the uncertainty category. Schedule C fails to surpass D in this case because of its poor showing in the raw-goodness component. Schedules A and B come in third and fourth, but for different reasons. Schedule A is very strong in the raw-goodness category so despite its poor showing in uncertainty, it is able to avoid last place. Schedule B, on the other had, has a very poor raw-goodness rating but also has little uncertainty in the quality and cost dimensions and thus does the best in the uncertainty category.

Consider another scenario where the meta slider for raw goodness is moved to zero and the threshold/limit and uncertainty slider each contribute 50% to the overall rating.

Schedule A: $rating = .88 * 0\% + 1 * 50\% + .17 * 50\% = .59$
Schedule B: $rating = .12 * 0\% + 0 * 50\% + 1 * 50\% = .5$
Schedule C: $rating = .33 * 0\% + 1 * 50\% + .5 * 50\% = .75$
Schedule D: $rating = .72 * 0\% + 1 * 50\% + .25 * 50\% = .63$

In this scenario, Schedule C takes the lead with D and A coming in distant second and third. This is simply because of C's strong showing in limits/thresholds and its strong showing in uncertainty. Without the raw-goodness category, A and D are not competitive with C. Schedule B again

comes in last, but not too far from the 3rd place A because of its perfect score in the uncertainty category.

For a final scenario, consider a case where the meta sliders are set such that raw goodness contributes 25% of the weight, thresholds/limits contribute 45% and uncertainty contributes 30%. In addition, let us decrease the quality slider in the raw goodness bank to a position equal with cost and duration, i.e., setting all to 33%.

Schedule A: $rating = .55 * 25\% + 1 * 45\% + .17 * 30\% = .64$
Schedule B: $rating = .39 * 25\% + 0 * 45\% + 1 * 30\% = .43$
Schedule C: $rating = .6 * 25\% + 1 * 45\% + .5 * 30\% = .75$
Schedule D: $rating = .58 * 25\% + 1 * 45\% + .25 * 30\% = .67$

In this last scenario Schedule C wins the prize. Because it has the least cost and low duration, it obtains a fair percentage of the available raw-goodness points. The better showing in raw-goodness, in conjunction with the emphasis on limits/thresholds and uncertainty give C the win.

To re-cap, Schedule C won two out of four scenarios and Schedules D and A each won one. The results of cases where one meta slider is set to 100%, and the others set to 0%, are obvious (assuming the initial weights of quality, cost, and duration in all the banks). If the raw goodness measure is the only concern, A wins. If the thresholds/limits component gets 100% of the weight, A, C, and D tie. In situations where uncertainty of cost and quality are the primary factors, B wins.

5 Gradual Utility Applied to Limits and Thresholds

In Sections 2, 3, and 4 we presented limits and thresholds as hard absolute values.[2] However, in many situations threshold/limit related utility may gradually *increase* as a threshold is approached or *decrease* as a limit is crossed. Consider the previous example where the cost limit is set to \$5.75. Say the individual in question has \$10 to spend, prefers to spend under \$5.75, but is willing to go as high as \$7.50. The previous presentation of hard thresholds, where cost-based limit utility changes immediately from 100% to 0% when cost crosses the \$5.75 limit fails to account for the individual's grudging willingness to spend more money if needed to obtain desirable results. Instead, to model this softer frugality, we must move to a model where utility gradually decreases after a limit is passed. For a quality perspective, consider a case where the individual desires quality over five but quality over two is still somewhat useful, quality over three a little more useful, and so forth. In this case, there is a gradual increase in utility from the point at which quality is two until it crosses the threshold at five, where quality-based threshold utility goes to 100%.

To account for models of gradually changing utility we must enhance the threshold/limit portion of our specification tool and the corresponding evaluation functions. Figure 2 illustrates the soft threshold specification mechanism applied to the situation described above. The conceptual tools pictured in the figure allow clients to specify gradual utility functions in a variety of ways. If a function is linear, a starting/ending point is specified along with a slope. If a function is non-linear (or linear) it can be described via its equation. If the client has an intuitive notion of the desired function, it can also be drawn graphically. Note that in the case of quality thresholds it is most natural to think of quality-based utility starting to increase at some client specified start point, perhaps zero, and then reaching full utility at the client specified 100% utility threshold. For cost

[2]Not to be confused with hard constraints. As illustrated by the examples in Section 4, schedules that violate the limits/thresholds can still be selected for execution. The thresholds/limits described in prior sections only denote points at which schedules fail to receive rewards from the points alloted to the limits/thresholds slider bank.

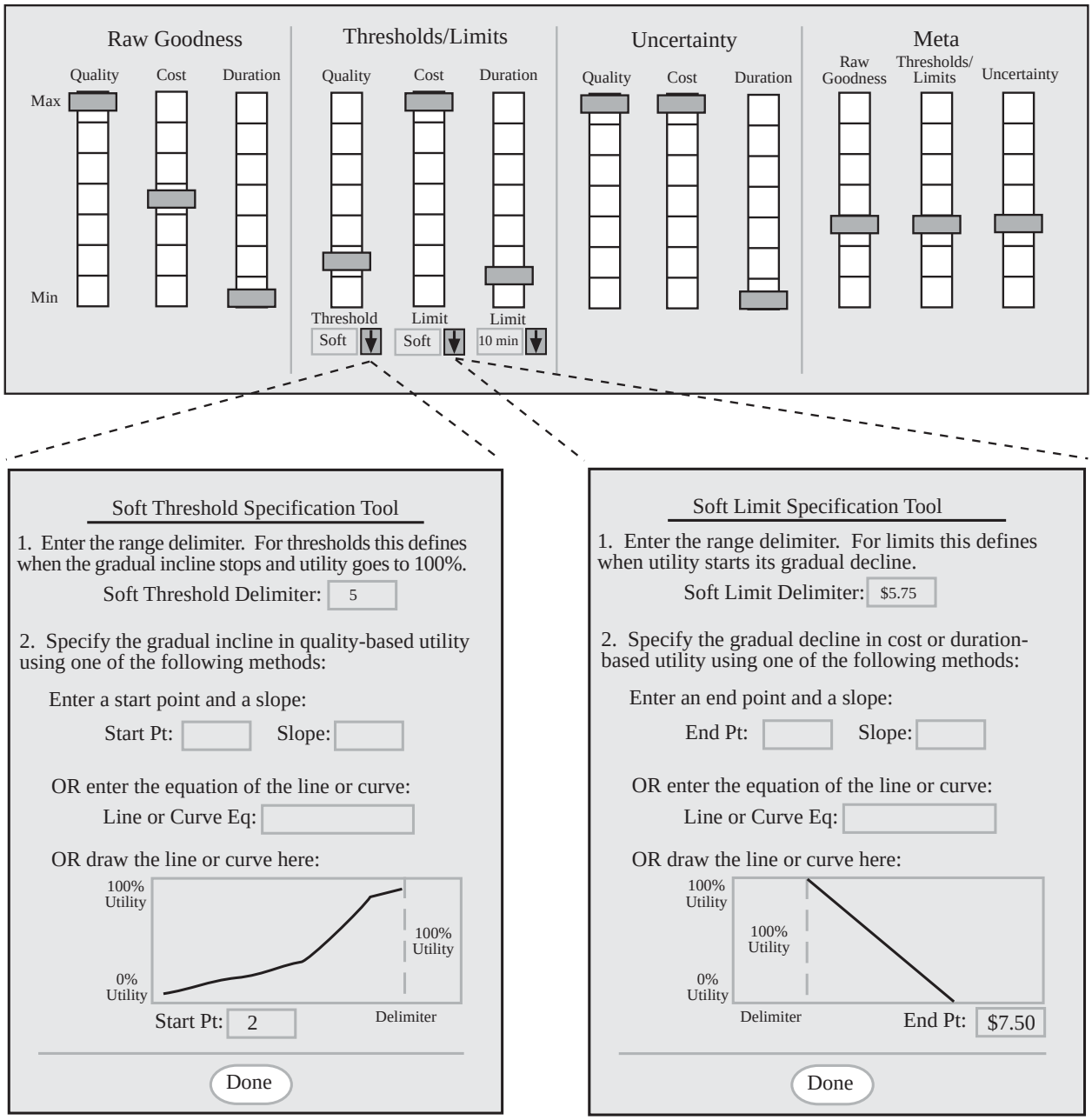

Figure 2: Slider Set and Gradual Utility Specification Tools

and duration limits, it is most natural to think of cost or duration-based utility starting to decrease after a limit is passed and eventually going to zero at some client specified end point.[3]

Integrating the gradual utility mechanism into the threshold/limit rating calculations is relatively straightforward. The equations in Section 3 pertaining to limits or thresholds remain as they are, but in situations where a soft threshold or limit is specified and the value is below the threshold or above the limit, the reward amount is defined by the curve, line, or function that describes the changing utility. Consider the quality case; if the quality value is under the client specified threshold, the gradual utility function defines how much utility, from 0% to 100%, quality at that level is worth and the resulting reward for the quality component is that utility value scaled by the percentage of points allocated to the quality slider for the limit/threshold bank. The general quality subcomponent equation follows:

$$
\begin{aligned}
&if\ \ (this.expected\_quality < client\_quality\_threshold)\ \ then \\
&\qquad rating_{quality} = utility\_function\_indexed\_by\_this.expected\_quality * \frac{quality\_slider\_value}{total\_points\_in\_threshold\_bank} \\
&elseif\ \ (this.expected\_quality \geq client\_quality\_threshold)\ \ then \\
&\qquad rating_{quality} = \frac{quality\_slider\_value}{total\_points\_in\_threshold\_bank}
\end{aligned}
$$

Obviously the entire conditional expression can be transformed into a single call to the utility function. In situations where quality values are below the start of the gradual increase in utility, they are given 0% utility ratings by the utility function. If quality values surpass the threshold between partial and full utility, they are given a 100% utility rating by the utility function. Values that fall in between these to ranges are given partial utility percentages as defined by the function. The other equations of the subcomponents are similarly modified.

Soft limits and hard limits are not mutually exclusive. Combinations of these approaches to dealing with limits and thresholds are possible even within the same criteria set. The appropriate equation is selected and applied when relevant. Relating the numerical ratings generated by a soft threshold to a rating generated by a hard threshold is also not an issue as each rewards on the same 0% to 100% scale and each is subsequently scaled by the percentage of allocated to the respective slider.

6 The Role of Hard Constraints and Negotiation

Hereto we have presented a satisficing view of limits and thresholds – where schedules may overrun cost and duration limits, or fail to achieve a certain quality threshold, and still be selected for execution. Our research focus is satisficing – determining a course of action from a set of alternatives, in an uncertain environment, that is most likely to generate the desired results. In our domain clients themselves do not typically have complete knowledge about what courses of action are possible, involving what trade-offs, even if they hand create the TÆMS task structure. The problem of evaluating all the trade-offs and building a "good" schedule is computationally infeasible even with a moderately complex task structure. Our satisficing approach to limits and thresholds allows clients to express what they *want*, not what they *know* is possible.

Hard constraints generated by circumstances such as "I have absolutely $5 to my name" have a place in the Design-to-Criteria paradigm. The question is what to do with the hard constraints? Should there be a separate specification mechanism for hard constraints on quality, cost and duration? Should the scheduler enforce the constraints by ruling out any schedule that does meet

[3]The word "gradual" is used to denote the notion of a soft threshold or limit, "gradual" does not mean that the slope of these functions must be non-zero or below some threshold. Step-wise functions and curves whose slopes oscillate between positive, negative, and zero are all valid utility functions.

them? The answer is not obvious. What if achieving a given set of hard constraints is not possible given the task model? We do not really know the answer to this last question until all promising alternatives are turned into schedules. What if the hard constraints can only be met by a schedule that does very poorly according to the evaluation criteria and relative to the other schedules? Is the poor schedule that meets the hard constraints better than a good one that fails in one or more respects? What if we have to do an exhaustive search though the space of $O(2^m)$ possible alternatives (where m is the number of actions) and build a schedule for each to find one that actually meets the hard constraints? What if we perform the search, and still do not find a schedule that meets the constraints? In this last case, the answer seems more clear. If the hard constraints cannot be met by any schedule, it only makes sense to apply the satisficing evaluation criteria to the schedules to determine the one that best meets the specifications, regardless of the unmet hard constraints. However, do we engage in that search in the first place?

This is an open issue in Design-to-Criteria scheduling. Currently the scheduler does support hard deadlines, but they are of a very different grainsize than typically thought about from the client perspective. Hard deadlines are associated with TÆMS tasks and methods and are specified via the TÆMS task model not via the criteria specification. These deadlines are enforced by the scheduler heuristics that put together schedules, not by the evaluation functions. The mapping from such fine grainsize deadlines to the evaluation criteria, and the role of these hard deadlines with respect to satisficing and the criteria specification is not clear. Should these constraints be softened in keeping with the satisficing notion of thresholds and limits presented in this paper? Should hard cost and quality requirements also be added to the lower-level scheduler internals? Or do these items belong in the evaluation criteria alone? While the grainsize questions require more detailed analysis, negotiation between the scheduler and its client seems to be the solution to questions about when to satisfice and the role of high-level hard quality, cost, and duration constraints.

Run-time interaction [4] between the client and the scheduler, or, from another viewpoint, interaction between the criteria specification mechanism and the evaluation functions could control and refine satisficing activities and handle problems involving hard constraints. Interactive negotiation during alternative evaluation could be used to refine client expectations based on what actually *seems* possible given the TÆMS model and the alternative abstraction. This would allow client and scheduler to cooperatively fine tune the criteria based on the estimated realities of the model before any work is spent building schedules. With the current model of criteria specification followed by application, it is possible that none of the generated schedules satisfactorily meet the client's ideal needs (though the one that best satisfices to meet the criteria will be returned). In this case, the client may prefer an alternate set of criteria rather than taking a satisficing view from the perspective of the original criteria. Negotiation during the scheduling phase could help refine the criteria based on what is actually expected of the schedules thus far produced at a given point. The refined criteria would then alter the selection of alternatives and through an iterative refinement mechanism the scheduler and the client could find the best satisficing solution. Negotiation during the scheduling process is clearly the next step in exploiting and leveraging the power of the Design-to-Criteria paradigm.

7 Conclusion

The evaluation functions and criteria specification mechanism presented here meet our current needs, with one exception. When the model was conceived, we viewed uncertainty as an attribute of quality, cost, and duration and accordingly designed a means to specify the importance of

uncertainty and uncertainty reduction. However, uncertainty is actually a satisficing/evaluation dimension in and of itself. Alternatives and schedules do not just have quality, cost, and duration characteristics that have some degree of certainty about them; alternatives and schedules have quality, cost, and duration characteristics and uncertainty about quality, uncertainty about cost, and uncertainty about duration. In other words, there are four different dimensions but they are represented using three models, the fourth dimension, uncertainty is represented by the probability distributions associated with quality, cost, and duration. The distinction is subtle and perhaps best illustrated by the feature that must be added to the criteria mechanism to make it complete. The criteria specification metaphor and the evaluation mechanism is missing a slider bank that defines *thresholds on uncertainty*, i.e., a slider bank that is directly analogous to the limits/thresholds bank for quality, cost, and duration. This bank would allow clients to specify that certainty is important only up to a particular threshold and then other criteria should dominate. Consider the following criteria setting:

- The raw-goodness quality slider is set to 100%.
- The proposed threshold-quality-certainty slider is set to 100% and a threshold value of 80% is provided.
- The meta slider for raw-goodness is set to 50% and the meta slider for certainty thresholds is set to 50%.

This setting specifies that certainty about quality is equally important with raw quality *until* quality reaches or exceeds the 80% certainty level. When this occurs, raw quality is the only factor used to differentiate between different schedules or different alternatives. Without the proposed certainty thresholds bank, clients cannot express this type of objective. The single uncertainty bank described in the previous sections correlates directly with the raw-goodness bank and without this proposed addition, there is not an uncertainty-centric bank that correlates with the existing limit/threshold bank.

Aside from this late revelation, we are pleased with the criteria mechanism presented here. It allows clients to reason about the relative importance of quality, cost, and duration in the context of raw goodness, thresholds or limits, and uncertainty. Evaluation function ratings in each case are proportional to the relative importances described by the client – differences in scale between quality, cost, and duration do not affect the rating process. The criteria specification process is intuitive and both powerful and simple. Client applications or clients specify only what is necessary to adequately describe their needs. As long as one slider in one slider bank contains values, the evaluation functions will work as advertised. Actually, even if the slider banks contain no value the evaluation functions will work properly. Empty slider banks denote a "no preference" criteria set and the evaluation functions will accordingly return an arbitrary selection.

Computational expense is not an issue with the new evaluation functions. The evaluation process is as inexpensive as it is effective. The $O(n)$, where n is the number of items being rated, cost factor is dominated at all times by other operations in the scheduler.

Finally, the evaluation functions and specification mechanism will support the addition of alternative generation methods that produce alternatives to address uncertainty in a pro-active manner. Where desired by the client, and so indicated by the evaluation criteria, alternatives of this type will be propagated, protected from pruning, and targeted for scheduling, assuming they best fit all the criteria constraints. The beauty of this entire approach is that the objective of meeting the desired criteria is ubiquitous throughout the scheduling process. Using evaluation functions as presented here, the scheduling system is truly design to specifications at every level.

References

[1] Keith S. Decker. TÆMS: A framework for analysis and design of coordination mechanisms. In G. O'Hare and N. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, chapter 16. Wiley Inter-Science, 1995. Forthcoming.

[2] Keith S. Decker and Victor R. Lesser. Quantitative modeling of complex computational task environments. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 217–224, Washington, July 1993.

[3] Keith S. Decker and Victor R. Lesser. Quantitative modeling of complex environments. *International Journal of Intelligent Systems in Accounting, Finance, and Management*, 2(4):215–234, December 1993. Special issue on "Mathematical and Computational Models of Organizations: Models and Characteristics of Agent Behavior".

[4] Alan Garvey, Keith Decker, and Victor Lesser. A negotiation-based interface between a real-time scheduler and a decision-maker. In *AAAI Workshop on Models of Conflict Management*, Seattle, 1994. Also UMASS CS TR–94–08.

[5] Alan Garvey and Victor Lesser. Design-to-time real-time scheduling. *IEEE Transactions on Systems, Man and Cybernetics*, 23(6):1491–1502, 1993.

[6] Alan Garvey and Victor Lesser. Design-to-time scheduling with uncertainty. CS Technical Report 95–03, University of Massachusetts, 1995.

[7] Alan Garvey and Victor Lesser. Representing and scheduling satisficing tasks. In Swaminathan Natarajan, editor, *Imprecise and Approximate Computation*, pages 23–34. Kluwer Academic Publishers, Norwell, MA, 1995.

[8] Thomas Wagner, Victor Lesser, and Alan Garvey. Design-to-criteria scheduling for intermittent processing. In *UMASS Department of Computer Science Technical Report TR-1996-81*, November, 1996.