Leveraging Uncertainty in Design-to-Criteria Scheduling *

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 1997-11

January 20, 1996

Abstract

Design-to-Criteria scheduling is the process of custom building a schedule to meet dynamic client goal criteria, using a task model that describes alternate ways to achieve tasks and subtasks. Formerly, Design-to-Criteria scheduling relied on simple expected value characterizations of method outcomes. The recent addition of uncertainty to the task model and its ubiquitous application in Design-to-Criteria scheduling has greatly improved four aspects of the scheduling process: *modeling* of tasks and task interactions, *evaluation* of schedules and schedule approximations, *focusing* of scheduling activities on more certain schedules when uncertainty reduction is important to the client, and *construction* of schedules that have more certainty and perhaps employ multiple ways to achieve a particular task to improve certainty. We describe the uncertainty representation and how it improves task models and the scheduling process, and provide empirical examples of uncertainty reduction in action.

*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

Representing and reasoning about uncertainty is one of the keys to scheduling computational structures in uncertain environments. This is particularly true when quality requirements and time and cost constraints are present. Additionally, with the inclusion of uncertainty modeling and propagation it is clear that there are many different dimensions and aspects of utility that can be used to evaluate the appropriateness of schedules. Consider the task of gathering information via the highly uncertain WWW to support a decision about the purchase of a statistical analysis software package. Certain clients may prefer a risky information gathering plan that has a potentially high pay-off in terms of information gathered, but also has a high probability of failure. Other, more risk averse clients might prefer a course of action that results in a lower pay-off in exchange for more certainty about the pay-off and a lower probability of failure.

Design-to-Criteria [3, 6] scheduling is the process of custom tailoring a way to achieve a high-level task via actions described in a TÆMS [2] model of the task, to fit a particular client's quality, cost, and duration criteria or needs. Recently the TÆMS task modeling framework was extended to model uncertainty about the quality, cost, and duration characteristics of tasks using discrete probability distributions. We have augmented and extended the Design-to-Criteria scheduling system to leverage this new explicit representation of uncertainty to build better custom schedules.

Uncertainty plays several roles in the Design-to-Criteria scheduling process. First, it enables the scheduler to represent and propagate uncertainty about tasks and their outcomes. This results in more accurate models of individual tasks, and more importantly, more accurate models of task sequences and task interactions. In contrast to reasoning from a single expected value, this enhancement supports notions like "30% of the time Task A will fail, 40% of the time it will generate fair results, and 30% of the time it will generate high-quality results." Because the models of tasks, task interactions, and sequences of tasks are more accurate, the scheduler builds better schedules.

The second role of uncertainty is in *evaluation;* it enables the scheduler to evaluate quality, cost, duration, *and* uncertainty trade-offs when custom building schedules to meet a particular client's needs. The addition of uncertainty to both the task model and goal criteria allows clients to specify how important, if at all, uncertainty reduction is relative to other schedule features like raw-goodness and threshold/limit specifications in each of the three modeled dimensions: quality, cost, and duration. Uncertainty's third role is in *focusing;* the scheduler uses the client's importance measure throughout the scheduling process to focus efforts on building schedules and partial schedules that best satisfice to meet the client's criteria. When uncertainty reduction is important, the scheduler may select tasks that have a high degree of certainty about the specified dimension(s) and trade-off utility in other dimensions as specified by the client's criteria. For example, if certainty in the quality dimension is important to the client relative to raw quality goodness, the scheduler may trade-off high quality for more certainty about quality when building schedules, resulting in schedules with lower overall quality but higher quality certainty. In situations where a deadline must be met, the scheduler may elect to trade-off quality or even short duration in exchange for certainty about duration, producing schedules whose durations are not as short as possible, but whose durations are more certain than the schedules that have the shortest durations. These simple examples are members of a large class of multi-dimensional attribute trade-offs that Design-to-Criteria considers when building schedules.

The fourth use of uncertainty in the scheduling process is in *construction*; when uncertainty is important to the client, the scheduler may take a more active approach to uncertainty reduction and elect to use more than one way of achieving various tasks in order to increase the certainty of results in the desired dimension(s). We discuss the use of uncertainty in the scheduling process in detail in Section 2 and demonstrate the power of uncertainty to produce better schedules in

Section 3.

This work falls into the general area of *flexible computation* [4], but differs from most flexible computation approaches in its use of multiple methods for task achievement (one exception is [5]), in its first class treatment of uncertainty, and in its ability to use uncertainty information in the selection of methods for execution. Much work in flexible computation makes use of *anytime algorithms* [1], algorithms that always have an answer at hand and produce higher quality results as they are given more time, up to a threshold. Our multiple methods approach can model any activity, including anytime algorithms, that can be characterized statistically and we place no constraints on the statistical behavior of the activities in question. In our work, uncertainty is a first class concept that both appears in the statistical descriptions of the available methods and is propagated and related as schedules and schedule approximations are generated. Unlike most work in anytime algorithms that focuses on the propagation of uncertainty[7], we can also include uncertainty and uncertainty reduction in the goal criteria and focus work on reducing uncertainty when important to the client. This ability stems from our task model's representation of alternative ways to perform various tasks. Because multiple-methods often exist to perform tasks, we can reason about the quality, cost, duration, and uncertainty trade-offs of different actions when determining which actions to perform, achieving the best possible overall results.

2 TÆMS and Design-to-Criteria Scheduling

TÆMS (Task Analysis, Environment Modeling, and Simulation) is a domain independent task modeling framework used to describe and reason about complex problem solving processes. TÆMS models serve as input to the Design-to-Criteria scheduler. TÆMS models are hierarchical abstractions of problem solving processes that describe alternative ways of accomplishing a desired goal; they represent major tasks and major decision points, interactions between tasks, and resource constraints but they do not describe the intimate details of each primitive action. All primitive actions in TÆMS, called *methods*, are statistically characterized in three dimensions: quality, cost and duration. Quality is a deliberately abstract domain-independent concept that describes the contribution of a particular action toward achieving the overall goal and the relative importance of its contribution. Thus, different applications have different notions of what corresponds to model quality. Duration describes the amount of time that the action modeled by the method will take to execute and cost describes the financial or opportunity cost inherent in performing the action. With the recent addition of uncertainty modeling, the statistical characteristics of the three dimensions are described via discrete probability distributions associated with each method.

To ground further discussion of scheduling TÆMS models, consider the simple information gathering task structure shown in Figure 1. The task structure models multiple different approaches for gathering information about WordPerfect via the WWW. A set of satisficing schedules produced by the Design-to-Criteria scheduler using four different sets of evaluation criteria is shown in Figure 1. Schedule A is constructed for a client interested in a fast, free, solution with any non-zero quality. Schedule B suits a client who wants a timely and free solution, but wants less uncertainty about the expected quality of the results. Schedule C is constructed for a user interested in a good quality, free, solution that can be obtained while she goes for a cup of coffee. Schedule D is generated to meet the criteria of a fourth individual who is willing to pay and wait for a high-quality response.

As demonstrated by this simple example, Design-to-Criteria scheduling is about custom building schedules to fit a particular client's criteria or needs. The two most important features of the Design-to-Criteria paradigm are the ability to reason about the quality, cost, duration, and uncertainty trade-offs of different solutions and partial solutions based on different goal criteria, and the ability

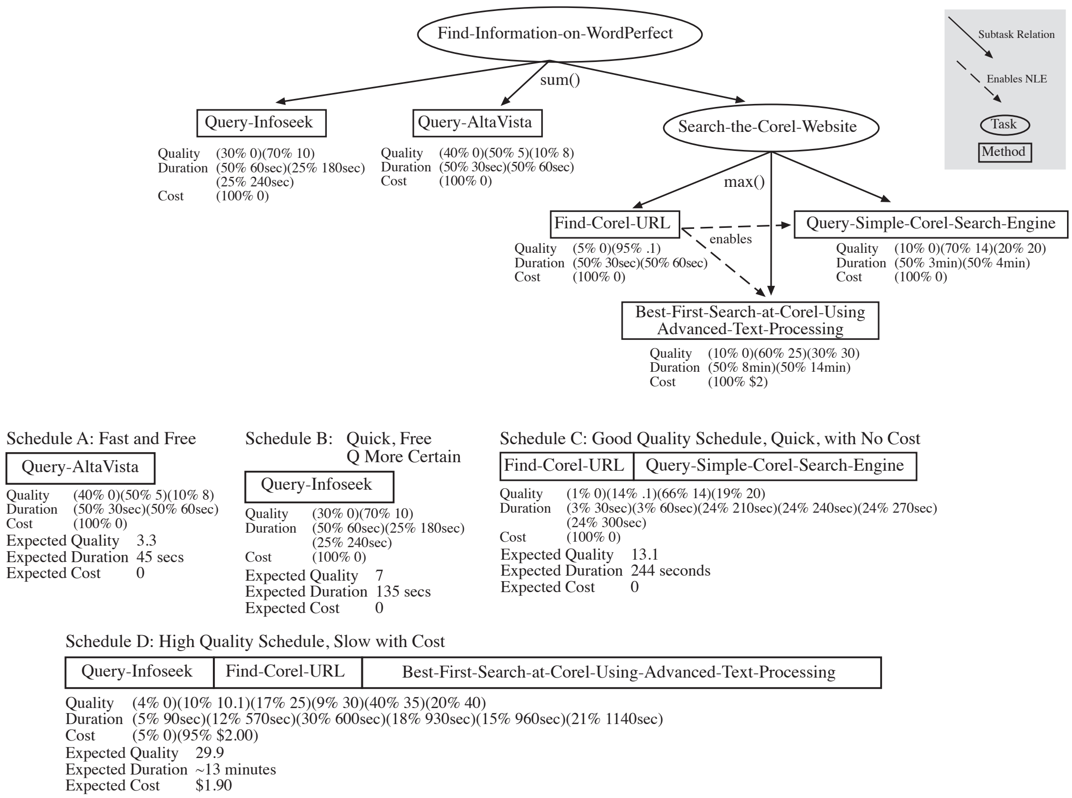

Figure 1: Info. Gathering Task Structure & Satisficing Schedules

use these utility attribute trade-offs to focus every step of the scheduling process. Satisficing also plays an important role in Design-to-Criteria. Satisficing in the scheduling process itself enables the scheduler to produce results when computational combinatorics preclude finding an optimal solution. Satisficing with respect to meeting the goal criteria allows the scheduler to produce a result that adheres to the spirit of the goal when the criteria cannot be satisfied optimally due to environmental and resource constraints.

Goal criteria are generated using a client specification metaphor called *sliders*. Sliders take on values from 0 to 100% and are arranged in slider banks where each bank contains a slider for quality, cost, and duration. The sliders in each bank sum to 100%. There are five banks in the current specification metaphor, each relating to a different class of concerns:

Raw Goodness This bank describes the relative importance of each dimension. For example, setting the quality slider to 50% and cost and duration to 25% expresses the notion that quality is twice as important as each of the other dimensions.

Certainty Whereas the set above expresses the relative importance of quality, cost, and duration, this set expresses the relative importance of certainty about quality, certainty about cost, and certainty about duration. Certainty about a particular dimension is the probability that the expected value or one better will result from execution. We discuss how certainty is calculated in greater detail below.

Threshold and Limits This bank allows the client to set limits and thresholds for quality, cost, and duration either using a fixed limit/threshold value or using a utility function that describes gradual changes in utility.

Certainty Thresholds This bank is analogous to the thresholds/limits bank above except that this bank focuses on the uncertainty associated with quality, cost, and duration. Schedules or alternatives whose certainty in a particular dimension meet or exceed the defined threshold are preferred. This enables clients to expression notions like "certainty in the quality dimension is not important as long as the schedule is at least 80% likely to produce the expected quality value or one better," as opposed to raw certainty objectives like "certainty in the quality dimension is important" that are expressed using the certainty bank.

Meta This slider set relates the importance of the four previous slider sets. This separation allows clients to focus on relating quality, cost and duration with each other in each of the classes above, then to "step back" and decide how important each of the different classes are relative to each other.

The incorporation of uncertainty into the criteria specification provides clients with a means to describe how important reducing uncertainty is for their application *relative* to raw-goodness and limits/thresholds. While the mapping from the sliders to utility functions is beyond the scope of this paper, it is necessary to describe how a particular dimension's uncertainty is computed. Certainty about a quality value is the probability that a quality equal-to or greater-than the expected value will result, i.e., the sum of the densities of all quality values in the quality distribution greater than or equal to the expected quality value for that particular distribution. The reason for this is semantic – more quality is always a good thing. Certainty about duration and cost is computed similarly, albeit that what is "good" is reversed – less cost and less duration are good things. Certainty about cost or duration is the probability that a value equal-to or less-than the expected value will result. It is important to note that the probabilities associated with expected values in all dimensions can be quite low as the distributions in question are not necessarily normal.

Unlike traditional scheduling tasks where the primary issue is how to order a particular set of methods, Design-to-Criteria must also consider the many possible combinations of alternative approaches for achieving the high-level task. Prior to the process of building schedules, the traditional method-ordering scheduling problem, the scheduler must enumerate the different ways that the high-level tasks can be achieved. Each "way" is a cheap to compute schedule approximation called an *alternative*. Alternatives contain unordered sets of primitive actions and estimates for the quality, cost, and duration distributions that would result from building a schedule from the alternative. Alternatives differ from schedules in that the ordering for the primitive actions has not yet been defined and the attribute estimates are computed without regard for complex task interactions. Alternatives are constructed bottom-up from the leaves of the task hierarchy to the top-level task node, i.e., the alternatives of a task are combinations of the alternatives for its sub-tasks.

The complexity of the alternative generation process is pronounced. A task structure with n methods leads to $O(2^n)$ possible alternatives at the root level. We control this combinatorial complexity by focusing alternative generation and propagation on alternatives that are most likely to result in schedules that meet the spirit of the client's goal criteria; alternatives that are less good at satisficing to meet the goal criteria are pruned from intermediate level alternative sets. For example, a criteria set denoting that certainty about quality is an important issue will result in the pruning of alternatives that have a relatively low degree of quality certainty.

After the alternative set for the high-level task is constructed, a subset of the alternatives are selected for scheduling. Again, complexity is the issue. For alternatives that have m methods, schedule construction via exhaustive search, $O(m!)$, is not feasible and even our polynomial heuristic approach precludes building schedules for all alternatives. Satisficing with respect to the client's goal criteria is used at this stage to select the alternatives that are most likely to lead to schedules that fit the criteria. As with alternative generation, if uncertainty is important to a particular client, schedules that reduce uncertainty in the desired dimensions will be produced.

Figure 2 illustrates the scheduler's ability to focus processing on the goal criteria at hand. The figure shows the root-level alternative sets generated for two different criteria specifications; one where raw quality is the only factor of importance and one where certainty about quality

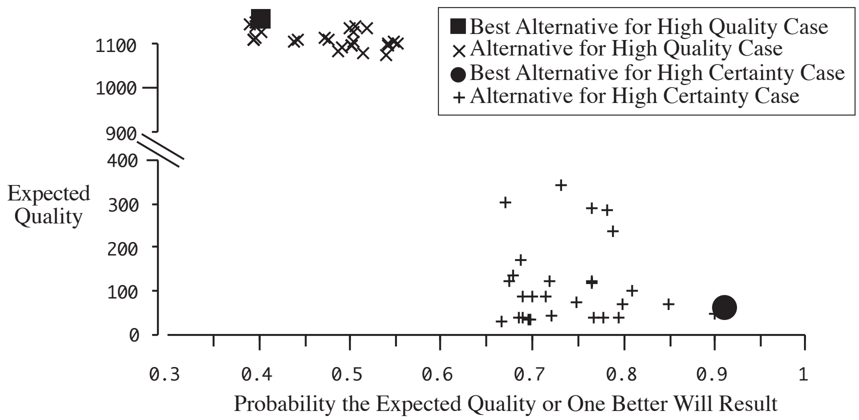

Figure 2: Alternatives Generated for Two Different Criteria Sets

is the only factor of importance. The task structure in question is moderately complex and has approximately 4×10^9 possible alternatives at the root level if focusing is not used to reduce the number of alternatives generated. When quality is the only factor, the alternatives generated have a high expected quality but also considerable quality uncertainty. In comparison, the alternatives generated for the the quality certainty case have lower expected quality but a much higher degree of certainty. The distributions are statistically significantly different in both the quality and quality certainty dimensions; one-tailed t-tests reject the null hypothesis of equivalence at the .05 level. If a third case where quality and quality certainty are equally important (omitted for clarity), was added to the figure the alternatives would fall partly in the quality only range and partly in the certainty only range; the overlap is due to the properties of the task structure where high quality methods tend to be uncertain and high certainty methods tend to have low quality. In this third case, the highest ranked alternative would be the same as the highest ranked in the certainty only case because it has the highest certainty to quality ratio.

In addition to its contribution to focusing the scheduling process, modeling uncertainty significantly improves the accuracy of task models resulting in better schedules. The simple task structure shown in Figure 3 illustrates this property. The distributions are simplified to make the point clear and although we will focus on the quality dimension, the same properties apply to cost and duration as well.

The *enables* arc between Task A and Task B denotes a hard precedence relationship – Task A must have quality before Task B can be performed. In other words, one of the methods for Task A, Method A1 or Method A2, must be executed before Method B1; any schedule that includes Method B1 must also include Method A1 or A2. The distributions associated with the methods are the *a priori* models of method execution and do not reflect the effect of the enables relation.

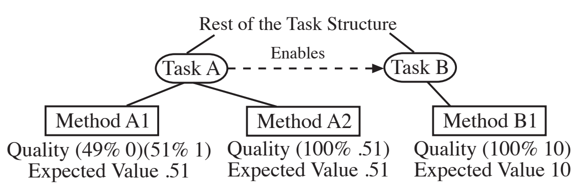

Figure 3: More Accurate Models Lead to Better Decisions

Consider a situation where the scheduler is comparing two schedules, one containing (A1,B1) and one containing (A2,B1). If the scheduler operates without a model of uncertainty the task model contains only the expected values associated with the methods. In this case, the expected value of both A1 and A2 is .51, but more importantly, is non-zero. From the scheduler's perspective, this means that the the enablement *always* occurs *regardless* of whether A1 or A2 is executed. Consequently Method B1 is always enabled regardless of which sequence, (A1,B1) or (A2,B1), is chosen and both sequences have an expected quality value of 10. In this case, the inaccurate expected value model of method outcomes hides information from the scheduler and can lead to poor choices, i.e., (A1,B1) and (A2,B1) are equally likely to be chosen if the durations and costs are identical.

However, when uncertainty is added to the model the uncertainty associated with the outcome of A1 is propagated and reflected in the expected results of (A1,B1). The 49% probability that Method A1 will fail to generate any results means that the required enablement relation occurs only 49% of the time. The resulting distribution of (A1,B1), (49% 0)(51% 10), reflects this uncertainty and the expected value of (A1,B1) is 5.1 rather than 10. For the (A2,B1) sequence the certainty about A2's outcome generates an appropriately certain quality result for (A2,B1), namely (100% 10), with an accordingly higher expected value of 10. With this improved model, even when uncertainty is not emphasized in the client's goal criteria, the scheduler makes better and more accurate scheduling decisions. In all cases (A2,B1) would be picked over (A1,B1) as long as all other factors were equal, i.e., assuming that A1 and A2 have similar durations and costs.

The scheduler can also take a more active role in uncertainty reduction by generating alternatives that contain more than one way (other alternatives) to achieve various tasks. This redundancy flavored scheduling may serve to reduce uncertainty and it provides the scheduler with more options to consider. This is critical in some situations involving hard deadlines because in the event of a failure there is not always enough time left to try a different solution approach, i.e., once committed to a course of action, it is sometimes too late to reschedule and try again if a failure occurs. Consider a brief example. Figure 4 shows a task structure fragment, the relevant method attributes, and two schedules. The results generated by Task A are necessary for Task B and there is a hard deadline of 30 minutes. Schedule 1 contains no redundancy, having one method for achieving Task A and one for achieving Task B. Schedule 2 contains redundant methods for achieving Task B and uses a lower quality but more certain and faster method for achieving Task A. If Schedule 1 is executed and method A1 fails, 20 minutes are wasted and there is not time to reschedule and execute method A2 followed by either B1 or B2 prior to the deadline. Additionally, if method B1 fails there is also not time to reschedule and execute B2. However, if Schedule 2 is executed, we are as certain as possible that some results will be generated by the deadline because A1 is very certain and the less-certain-but-higher-quality B1 is followed by the more-certain-but-lower-quality B2. Considering uncertainty in conjunction with redundancies is clearly important in some situations. When the redundancy alternative generation feature is used, the alternatives that contain redundant activities are added to the alternative set and compared to the goal criteria in the same fashion as the non-redundant alternatives. Thus, the scheduler continues to focus processing on alternatives that best satisfice to meet the overall goal criteria – uncertainty does not dominate the evaluation mechanism unless so specified by the goal criteria.

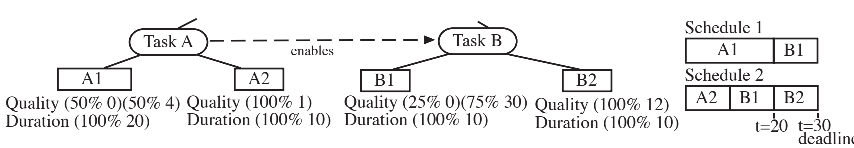

Figure 4: Redundancy Can Be Critical

Modeling uncertainty improves and empowers other aspects of the scheduling process as well. In environments where rescheduling is undesirable the scheduler can use the probability distributions to design more fault tolerant schedules. For instance, if fault tolerance with respect to duration is desired, the scheduler can build schedules by estimating method execution times using the 95th percentile duration value rather than the expected value. In this situation, uncertainty about finish times still gets propagated throughout the schedule, but timing assumptions are based on a higher value that is by definition very certain.

The uncertainty representation can also improve the probability that little work is wasted in the event of a mid-schedule failure. Because of task interactions it is possible that a method failure anywhere in the schedule can void all the work done up to that point. Modeling uncertainty makes it possible for the scheduler to move the highly uncertain activities toward the front of the schedule, thus reducing the likelihood of doing work that is voided later in the schedule. We are investigating integrating this concept with the other method rating heuristics that build schedules from alternatives.

3 Demonstrating the Power of Uncertainty

To illustrate the type of leverage provided by an explicit model of uncertainty, let us consider the problem of custom building schedules for two different clients from a moderately complex task structure. The task structure has methods that fall into three general categories. 1) Methods that have high expected quality values also tend to take longer and are highly uncertain in both the quality and duration dimensions. 2) Methods that have low expected quality also tend to take less time to execute and are more certain in both the quality and duration dimensions. 3) Methods that have medium expected quality also take a moderate time to execute and are moderately certain.

The high-quality-but-uncertain methods model information gathering tasks that are risky but also have a probability of a large information pay-off. For example, methods of this type may find information about a software product by submitting multiple queries to Infoseek and Altavista, going to the URLs, retrieving multiple documents from each site, and processing them. As the information located can range from useful new information with wide-scale ramifications to utterly useless information that is not relevant, there is the probability of big pay-offs and also the probability of zero or poor results. Since methods of this type use a large amount of active web search on sites that are unknown *a priori*, predicted duration is also long and uncertain. The low-quality-but-more-certain methods model information gathering tasks where information is retrieved from individual sites that are known and modeled. Since the information is predicted to be fairly narrow in scope, these methods lack the potential for big pay-offs, however, since the methods search only one site and the site in question is modeled, durations are short and fairly certain. The middle-quality-middle-certainty methods employ combinations of these behaviors.

Since the first client, Client A, is planning other activities based on the predicted outcome of schedule execution, this client is interested in both schedule raw-goodness and schedule certainty. In the raw-goodness slider bank the quality slider is set to 75% and the duration slider set to 25%, i.e., overall quality is 3 times more important than overall duration. In the certainty bank the quality and duration sliders are each set to 50%, meaning that certainty about the estimated quality and certainty about the estimated duration are equally important. The meta slider for raw-goodness is set to 40% and the meta slider for certainty is set to 60%, denoting that uncertainty reduction is 1.5 times more important than raw schedule goodness. Unlike Client A, Client B has much simpler needs and is only interested in raw-goodness. As with Client A, the raw-goodness quality slider for this client is set to 75% and the raw goodness duration slider is set to 25%. The meta-slider for raw goodness is set to 100% denoting that raw goodness is the only issue of importance to this client.

Figure 5 shows the expected quality and expected duration of the top-level alternatives generated for Clients A and B; intermediate alternative sets were pruned according to the client's goal criteria as discussed in Section 2. Despite both clients setting the raw quality and duration sliders to the same values, Client B's alternatives always have higher expected quality and higher expected duration than Client A's. Since neither client is using hard deadlines, this is attributable to Client A's emphasis on certainty about quality and certainty about duration. Figure 6 tells the rest of the story. As Client A put 60% of the overall weight on certainty in the quality and duration dimensions, the alternatives generated for Client A trade-off between raw quality, raw duration, quality certainty, and duration certainty, rather than just trading-off quality and duration. Figure 6 also shows the price of B's high expected quality – the expected values are also predicted to be much more uncertain than those of Client A.

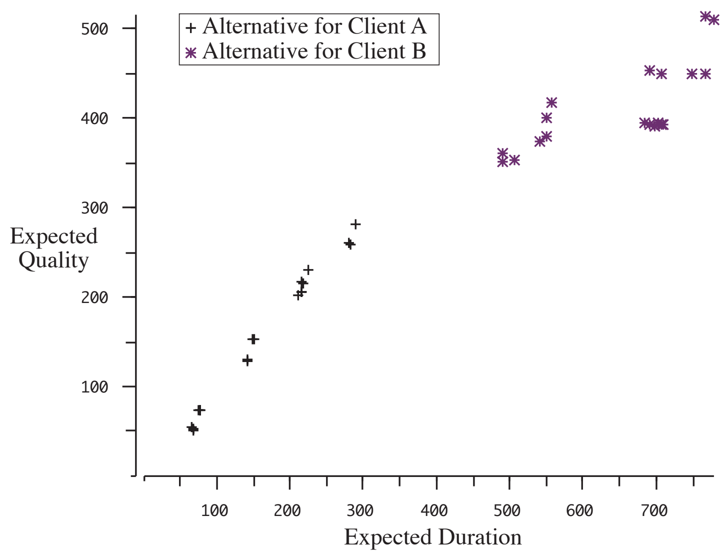

Figure 5: Alternatives for A and B

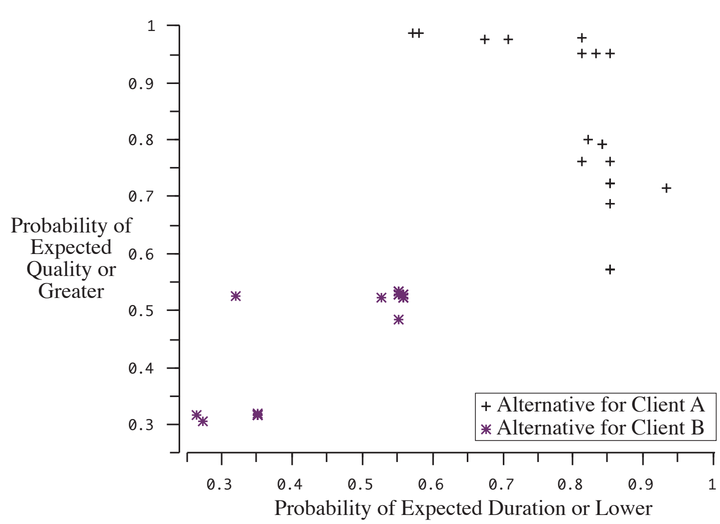

Figure 6: Probability of Expected Values of Alternatives

The quality and duration attributes of the schedules produced from a subset of these alternatives are similar to the attributes of the alternatives. In this case, the estimates contained in the alternatives are fairly good indicators of the schedules produced from the alternatives. This indicates that subtask interactions in the alternatives generated and targeted for scheduling were fairly simple and generally involved hard-precedence constraints. In keeping with intuitions, the highest rated schedule for Client B is that which has the highest expected quality with respect to duration. However, Client A's "best schedule" has a reasonably good quality for its expected duration and a high degree of certainty about its expected quality and duration values.

The quality and duration results of executing the best schedules for each client thirty times are shown in Figure 7. Whereas Client A's executions produced a tightly spaced set of quality and duration values, Client B's highly uncertain schedule produced a wide range of results. Of the thirty runs, Client A's results meet or beat expectations in the quality dimension 90% of the time, in the duration dimension 50% of the time, and in both the quality and duration dimensions 50% of the time. In contrast, Client B's results only meet or beat quality expectations 63% of the time, duration expectations 16% of the time, and both dimensions combined 13% of the time. Additionally, the uncertainty in B's quality dimension incurred more rescheduling because of methods failing to return any results (problematic because of task interactions). On average, B's plan required scheduling 2.1 times per each execution, with a variance of .71, whereas A's only required 1.2 schedulings on average with a variance of .21. The 25% trimmed mean brings out the contrast even more – B's scheduling average remains 2.1 but A's 25% trimmed mean drops to 1.0, denoting no rescheduling during execution.

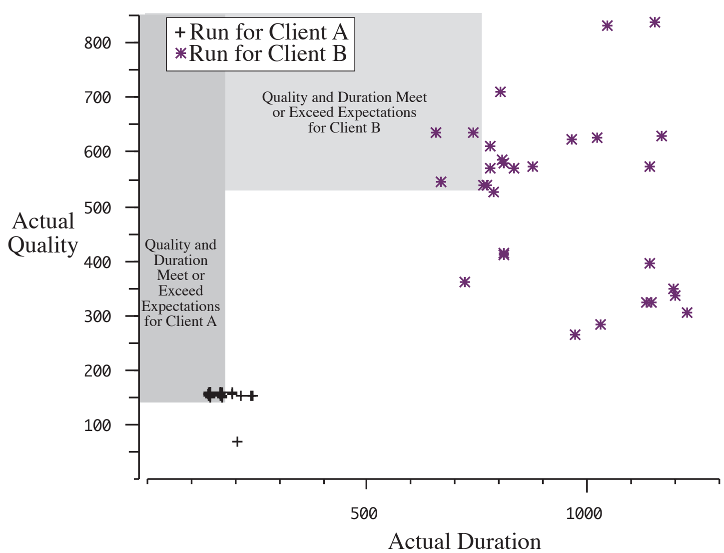

Figure 7: Execution Results for A and B

4 The Future Role of Uncertainty

As discussed in Section 2, the addition of uncertainty to the TÆMS modeling framework increases the accuracy of TÆMS models. The uncertainty enhancement is leveraged ubiquitously by Design-to-Criteria scheduling to better statistically reason about task interactions, to produce schedules that more fully satisfice to meet client's needs, and to improve the efficiency of the scheduling process. We have discussed these issues and demonstrated the power of using uncertainty in Design-to-Criteria scheduling. The issues of modeling improvement, measuring uncertainty, reasoning about attribute trade-offs with uncertainty, and working to reduce uncertainty are all applicable to research beyond Design-to-Criteria scheduling.

One area of future uncertainty-related work in Design-to-Criteria scheduling involves moving uncertain actions toward the front of the schedule to reduce the amount of work that is potentially wasted by an action failure. Because of complex task interactions and the complex semantics of the functions that determine how quality is accumulated by tasks via their subtasks, determining which actions are most important to the schedule and to what degree, is not a trivial or computationally cheap process. Contingency scheduling for methods likely to fail is also a possibility. Other future efforts in Design-to-Criteria will center around negotiation between the scheduler and its clients, which may be other AI problem solvers or humans. Negotiation during the scheduling process can iteratively refine client goal criteria based on what is actually being produced by the scheduler. This is important because often if the scheduler cannot produce schedules that satisfice *well enough* with respect to the goal criteria, due to task limitations or resource constraints, the client may prefer to submit a different set of goal criteria and try again, exploring the solution space prior to selecting a course of action.

References

[1] T. Dean and M. Boddy. An analysis of time-dependent planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 49–54, St. Paul, Minnesota, August 1988.

[2] 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.

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

[4] Eric Horvitz, Gregory Cooper, and David Heckerman. Reflection and action under scarce resources: Theoretical principles and empirical study. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, August 1989.

[5] Eric Horvitz and Jed Lengyel. Flexible rendering of 3d graphics under varying resources: Issues and directions. In *Proceedings of the AAAI Symposium on Flexible Computation in Intelligent Systems*, Cambridge, Massachusetts, November 1996.

[6] Thomas Wagner, Alan Garvey, and Victor Lesser. Satisficing evaluation functions: The heart of the new design-to-criteria paradigm. In *UMASS Department of Computer Science Technical Report TR-1996-82*, November, 1996.

[7] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996.