Dynamic Construction and Refinement of Utility-Based Categorization Models

Kim Leng Poh
Department of Industrial & Systems Engineering
National University of Singapore
Kent Ridge, Singapore 0511

Michael R. Fehling
Laboratory for Intelligent Systems
Department of Engineering-Economic Systems
Stanford University, CA 94305-4025

Eric J. Horvitz
Decision Theory Group
Microsoft Research
Redmond, WA 98052-6399

June 2, 1999

Key words:
Model construction, categorization, knowledge representation, control of reasoning

Abstract

The actions taken by an automated decision-making agent can be enhanced by including mechanisms that enable the agent to categorize concepts effectively. We pose a utility-based approach to categorization based on the idea that categorization should be carried out in the service of action. The choice of concepts made by a decision maker is critical in the effective selection of actions under resource constraints. This perspective is in contrast to classical and similarity-based approaches which seek completeness in concept description with respect to shared properties rather than the effectiveness of decision making. We propose a decision-theoretic framework for utility-based categorization which involves reasoning about alternative categorization models consisting of sets of interrelated concepts at varying levels of abstraction. Categorization models that are too abstract may overlook details that are critical for selecting the most appropriate actions. Categorization models that are too detailed, however, may be too expensive to process and may contain information that is irrelevant for selecting the best action. Categorization models are therefore evaluated on the basis of the expected value of their recommended action, taking into account the associated resource cost required for their evaluation. A knowledge representation scheme, known as *probabilistic conceptual networks*, has been developed to support the dynamic construction of models at varying levels of abstraction. This knowledge representation scheme combines the formalisms of influence diagrams from decision analysis and inheritance/abstraction hierarchies from artificial intelligence. We also propose an incremental approach to categorical reasoning which involves the dynamic construction and refinement of categorization models. A model may be improved by making the concepts under consideration either more abstract or more detailed. The expected increase in value of the recommended action may be used to direct and control the direction of model improvements. By applying decision-theoretic control of model refinement, a resource-constrained actor iteratively decides between continuing to improve the current level of abstraction in the model, or to act immediately.

I. Introduction

The ability to dynamically categorize distinctions into useful concepts and use these concepts effectively is a core competency of intelligent decision making. The *utility-based* approach to categorization is founded on the idea that categorization is fundamentally in service of action; the choice of concepts and the level of generality of concepts employed by an actor plays a critical role in the selection of appropriate actions [1]. This approach differs from classical and similarity-based approaches [2] to classification which seek logical *completeness* in concept description in terms of sets of attributes, defined by sensory inputs, rather than by considering the action-oriented *effectiveness* of categories.

A resource-constrained agent relies on its ability to interpret sensor observations to maintain a plausible representation of conditions in a dynamically changing world. Decisions are typically based on beliefs that are computed about distinctions that are not directly observed. The agent rely on a set of previously encoded distinctions about hidden and observed states of the world and knowledge about probabilistic dependencies among these distinctions to make inferences about events, objects, and conditions. The literature on decision making under uncertainty largely has addressed methods for computing beliefs and taking ideal actions. There has been little work on processes for reasoning about the nature of the concepts that are used in a decision model. Moving beyond consideration of means of computing the best decision to make given a set of observations and a static model, we have investigated an important modeling decision—selecting the best categories to use in representing the world. An actor with the ability to make such a choice can dynamically build, refine, and solve a *categorization decision model*, a decision model for representing alternative categories for classifying sensor information together with alternative courses of action. Building and solving categorization decision models requires decision making about the level of abstaction with which to consider key distinctions or groups of distinctions. Proper choice of a level of abstraction may be crucial to an actor whose further deliberations and actions are constrained by limited computational resources.

Introducing model abstractions through selective categorization introduces tractability of decision making inference at the expense of decision quality. Categorization of distinctions creates generalizations that hide potentially relevant details in a decision model. If computational and representational resources were free or inexpensive, there would be little reason to remove detail though categorization. Under conditions of limited resources, however, an actor may find that representing objects and events in the world at too detailed a level may require the subsequent expenditure of intolerable computational time in computing optimal decisions. A classification that is too detailed may contain information that is irrelevant to the choice of action, thus forcing the actor to waste cognitive effort without gain. On the other hand, a categorization model that is too abstract may overlook details that significantly diminishes the expected value of selected actions. When constructing and refining a categorization decision model, a resource-constrained actor can enhance its decisions by expending some resource to consider the tradeoff between the expected benefit of using more detailed models to increase the value of action and the resource costs entailed by computing decisions with a more detailed model.

We have developed a decision-theoretic approach to utility-based categorization as schematized in Figure 1. Our framework elucidates the deliberative processes that underlie an intelligent actor's utility-maximizing choice of actions. The influence diagram in the middle of Figure 1 depicts an intelligent actor deliberating about the choice of a *conceptual cover* based on observed sensor information. A conceptual cover is a set of mutually exclusive and exhaustive concepts at varying levels of abstraction that conceptually accounts for key aspects of the current situation. We consider the task of determining a best conceptual cover for a decision problem. For each conceptual cover being considered, a corresponding categorization decision model may be constructed, as shown by the series of influence diagrams at the bottom of Figure 1. The optimal choice of categorization decision model is made by identifying the model that maximizes a utility function which takes into account the cost associated with solving the categorization model in addition to the value of the action recommended by the model.

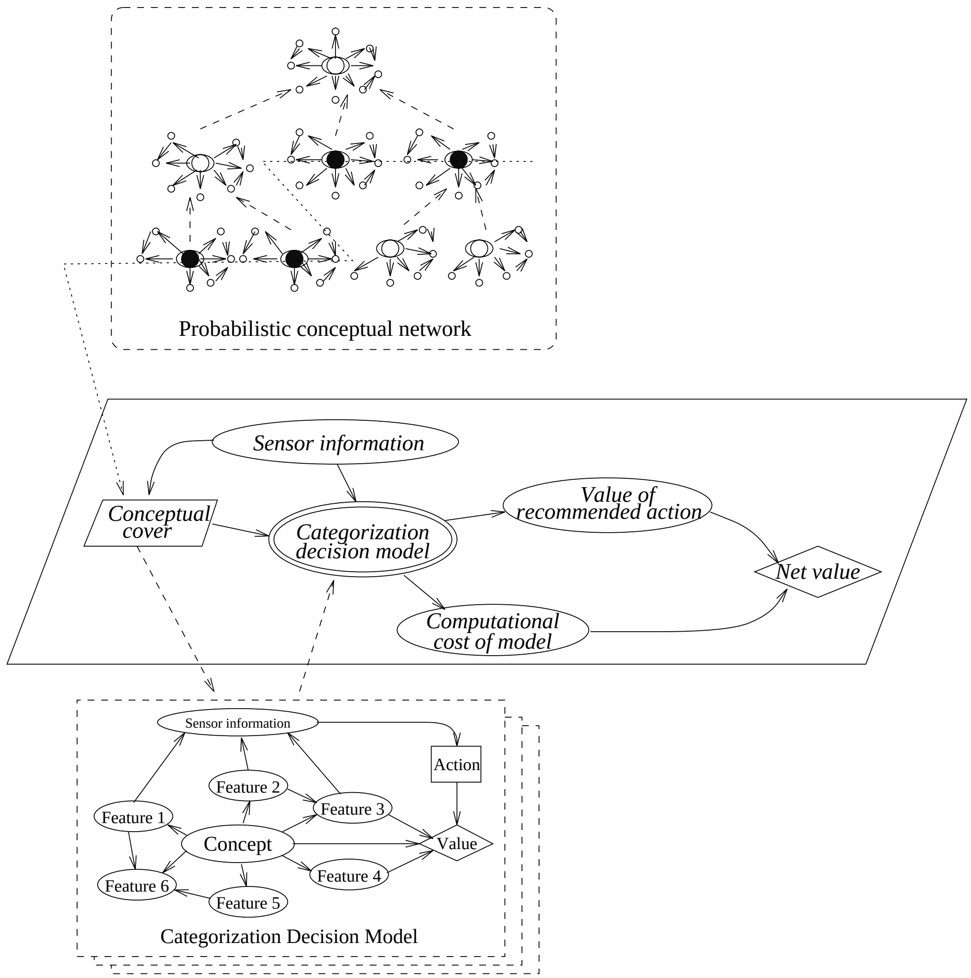

Figure 1: A decision-theoretic framework for utility-based categorization.

Domain knowledge for utility-based categorization is represented with a *probabilistic conceptual network* (pc-net) as shown at the top of Figure 1. A pc-net is a knowledge representation which combines the formalisms of influence diagrams from decision analysis and inheritance/abstraction hierarchies from artificial intelligence. In pc-nets, a concept is represented by a special form of probabilistic influence diagram we term a *probabilistic concept diagram* (pc-diagram). The pc-diagrams are in turn connected in a subsumption hierarchy such that concepts higher up the hierarchy subsume the concepts lower in the hierarchy.

Pc-nets allow categorization decision models to be dynamically constructed at varying levels of abstraction. A level of abstraction for the model is characterized by a conceptual cover. Concepts included in a conceptual cover are chosen from the pc-net at various levels in the concept hierarchy. The lower the level from which the concepts are obtained, the more detailed the conceptual cover will be. Conversely, the higher the level from which the concepts are obtained, the more general or abstract the level of description will be. The construction of the categorization decision model is performed by retrieving only those pc-diagrams for the concepts in the conceptual cover, and performing simple graphical manipulations on the pc-diagrams. We propose an incremental approach to categorical reasoning whereby an actor may iteratively improve a categorization model that is being constructed by considering the trade-off between the cost of additional model improvement and the improvement in the expected value of action recommended by the model.

This paper is organized as follows: In Section II, we describe an automated machining application which will be used to illustrate our approach. Section III describes the probabilistic conceptual network knowledge representation scheme. Section IV describes the use of categorization decision models and formalizes the procedures for model construction. Section V introduces the incremental approach to model improvement. Section VI presents the idea of control of model-refinement. Section VII shows the application of our approach in guiding the automated machine under different situations and context. Section VIII discusses some research issues related to this research. Finally, we conclude in Section IX by describing research we intend to pursue in the future.

II. An Automated Machining Application

We will illustrate the dynamic construction and refinement of categorization models in utility-based categorization with a real-world example of an automated machining problem. This is similar to an application described in [3]. Automated machining operations are important parts of any intelligent manufacturing system. They require the automation of the human operator's efforts to monitor and make appropriate adjustments to the state of the machine. An automated machining system typically has sensors which acquire data on (1) dimensions of the workpiece, (2) acoustic emission from the machining processes, (3) cutting forces (dynamometer readings), and (4) electric current (ammeter), etc. These observations are then used to determine the state of the machine and appropriate actions are taken to ensure the continuous operation of the plant so as to minimize production cost. The possible states of the machining process at various level of abstraction are illustrated in Figure 2. Table 1 lists the features characterizing these states of the machine.

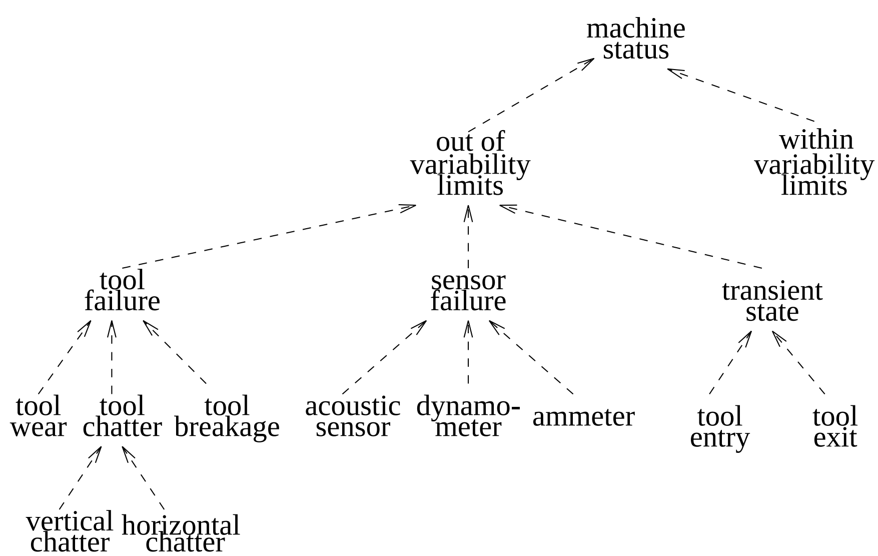

Figure 2: Hierarchy for states of a an automated milling machine.

Table 1: Descriptions of observed features.

Feature	**Description**
AE-mag	acoustic emission magnitude
ΔAE-mag	change in acoustic emission magnitude
AE-freq	acoustic emission frequency
dyn-freq-x	cutting force frequency in x-direction
dyn-freq-y	cutting force frequency in y-direction
AE-mean	mean of the acoustic signal
ΔAE-mean	change in the mean of the acoustic signal
dyn-rms-x	cutting force in the x-direction
Δdyn-rms-x	change in cutting force in the x-direction
dyn-rms-y	cutting force in the y-direction
Δdyn-rms-y	change in cutting force in the y-direction
AE-peak	acoustic emission peak value
dyn-peak-x	peak cutting force in x-direction
dyn-peak-y	peak cutting force in y-direction
current	motor current

At the most abstract level, the state of the machining process is either "within variability limits" or "out of variability limits." The event concept "out of variability limits" may be refined into three alternative subconcepts: "tool failure," "sensor failure," or "transient state." The latter occurs during the cutting tool's entry into or exit from the workpiece. "Tool chatter" is another subconcept of "tool failure" which is typically characterized by an event in which an acoustic emission signal increases dramatically in amplitude as does the frequency content of the dynamometer. If left unchecked, tool chatter can result in tool, workpiece or machine damage. Remedies for this problem include reducing the depth of cut or reducing the feed rate. "Tool wear" is typically characterized by a gradual increase in acoustic emissions, and by a gradual increase in cutting force as measured by the dynamometer. A tool that is worn out needs to be resharpened or replaced in order to achieve the desired surface finish and dimensional tolerances. "Tool breakage" is typically characterized by an acoustic emission exhibiting a high amplitude peak at the moment of tool fracture, and followed by a sharp drop in signal amplitude to a level below that of normal. It is also characterized by a steep rise in cutting forces, followed by a drop before finally continuing at a value above the average. Tools that are broken cannot perform any machining task and must be replaced immediately.

III. Probabilistic Conceptual Network Representation

A *probabilistic conceptual network* (pc-net) is a knowledge representation scheme designed for reasoning about concepts and categorical abstractions in utility-based categorization. The scheme combines the formalisms of abstraction and inheritance hierarchies from artificial intelligence, and probabilistic networks from decision analysis. It provides a common framework for representing conceptual knowledge, hierarchical knowledge, and uncertainty. It facilitates dynamic construction of categorization

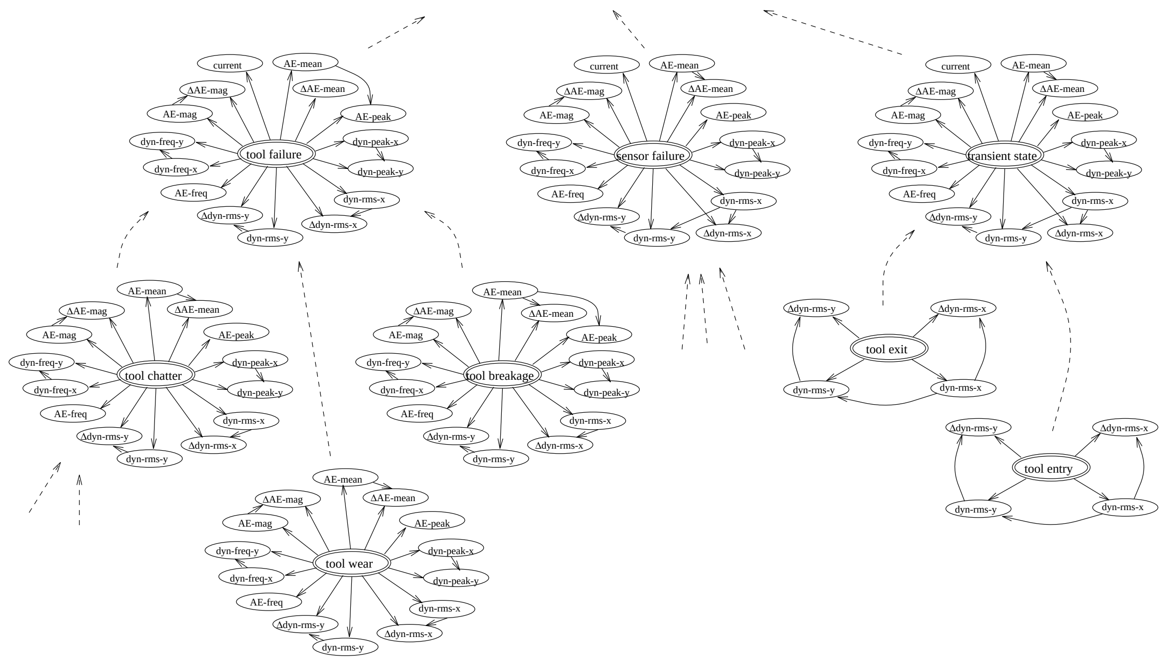

Figure 3: A fragment of the pc-net for the automated machining problem

decision models at varying levels of abstraction.

A *pc-net* consists of a *probabilistic concept hierarchy* (*pc-hierarchy*) connecting a set of *probabilistic concept diagrams* (*pc-diagrams*). Each node in the pc-hierarchy represents a concept, and the links in the hierarchy specify subsumption relations among the concepts thereby organizing the concepts at various level of abstraction or specificity. In the current version of pc-net, we will consider only single-parent hierarchy in which the pc-hierarchy is a tree. Each concept within the pc-hierarchy is represented individually by a pc-diagram. We may visualize a pc-net as a hierarchical organization of pc-diagrams. Figure 3 displays a fragment of the full pc-net used by the machining actor in our example, which we shall call T1000.

Figure 4 shows the pc-diagram for the concept "tool chatter" used by T1000. A pc-diagram for a concept is a special probabilistic influence diagram[1] representing

[1]A probabilistic influence diagram is an influence diagram with only probabilistic nodes and

knowledge about the probabilistic relations between the concept and the features that characterize it. As a convention, we direct arcs from the concept to its feature nodes. For each feature node F in the pc-diagram for concept C, we store a probability distribution[2] of the form

$$p(F|C, B^C(F))$$

where $B^C(F)$ is the set of conditional predecessors of F, excluding C (if any). For example, in the concept "tool chatter," a probability distribution for

$$p(\Delta\text{AE-mag}|\text{"tool chatter"}, \text{AE-mag})$$

is assessed. In the pc-diagram for C, we always represent C as a deterministic node because we do not need the distribution $p(F|\neg C, B^C(F))$. Arcs between feature nodes indicate possible relevance or probabilistic dependency among the features given the concept concerned. For example, the arc between the node "AE-mag" and the node "ΔAE-mag" indicates that information about the current magnitude of acoustic emission may provides information about the change in magnitude of acoustic emission. The direction of this arc could be reversed without any change in assertion about possible dependency.

Suppose concept C_i is a subconcept of C_j, denoted $C_i \ll C_j$. We define the *subsumption probability* of C_i given C_j to be the conditional probability $p(C_i|C_j)$, i.e., the probability that a subconcept C_i of C_j is true given that the concept C_j is true. We may rewrite the subsumption probability $p(C_i|C_j)$ as $\frac{p(C_i \wedge C_j)}{p(C_j)}$. However, $C_i \ll C_j$ implies that $p(C_i \wedge C_j) = p(C_i)$. Therefore

$$p(C_i|C_j) = \frac{p(C_i)}{p(C_j)}. \tag{1}$$

In other words, the subsumption probability is simply the ratio of the prior probabilities of the concepts.

conditioning arcs. They are also called Bayesian networks or probabilistic networks.

[2]We shall assume that background information ξ is used in all the probability distributions.

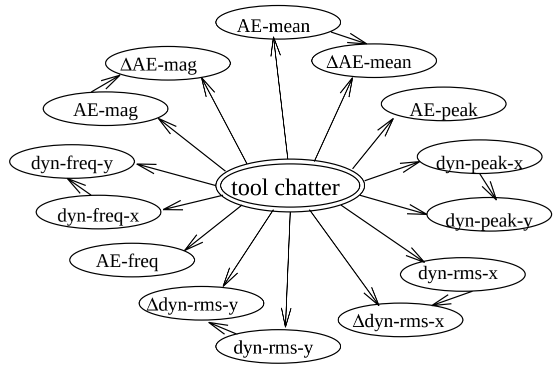

Figure 4: The pc-diagram for "tool chatter."

We say that two pc-diagrams are *graphically consistent* if their graphical union does not produce a cycle [1]. Given the special graphical structure imposed on pc-diagrams, any inconsistency must be based in the directions of the arcs between the features. Hence, making a set of pc-diagrams graphically consistent involves only the reversal of arcs between the features. Unless otherwise stated, we shall assume that all the pc-diagrams in a pc-net are graphically consistent with each other.

Let $\{C_1, \ldots, C_n\}$ be a set of mutually exclusive concepts whose pc-diagrams are already assessed. Suppose we wish to generalize these n concepts into a single super-concept S, then we may relate the probability distributions in the pc-diagram for S to those in the pc-diagrams for the subconcepts. In the pc-diagram for S, we have

$$B^S(F) = \cup_{i=1}^n B^{C_i}(F) \tag{2}$$

for all features F, and

$$\begin{aligned} p(F|S, B^S(F)) &= \sum_{i=1}^n p(F|S, B^S(F), C_i) p(C_i|S, B^S(F)) \\ &= \sum_{i=1}^n p(F|C_i, S, B^{C_i}(F), B^S(F) \setminus B^{C_i}(F)) \frac{p(C_i|S) p(B^S(F)|C_i, S)}{p(B^S(F)|S)} \end{aligned}$$

where $p(C_i|S)$ is the subsumption probability for concept C_i given the superconcept

S. Using the fact that for each i, C_i logically implies S since $C_i \ll S$, and that $B^S(F) \setminus B^{C_i}(F)$ is conditionally independent of F given C_i and $B^{C_i}(F)$,[3] we have

$$p(F|S, B^S(F)) = \sum_{i=1}^{n} p(F|C_i, B^{C_i}(F))p(C_i|S)\frac{p(B^S(F)|C_i)}{p(B^S(F)|S)} \tag{3}$$

For those features which do not have any feature predecessor, Equation (3) simplifies to

$$p(F|S) = \sum_{i=1}^{n} p(F|C_i)p(C_i|S) \tag{4}$$

In practice, to build the pc-diagram for S from those of its subconcepts, we would first use Equation (4) to evaluate the probability distributions for those features which do not have any feature predecessor. We then proceed, using Equation (3), to the other features in an order which is consistent with the partial ordering induced by the pc-diagram for S.

Our pc-net formalism uses an inheritance mechanism whereby a concept may share information about features from a concept higher up the hierarchy. It does so by taking advantage of a form of conditional independence called *subconcept independence*[4] which is not conveniently represented with ordinary influence diagram. A feature is said to be subconcept independent of a concept if knowledge about the feature does not affect the actor's belief about any of that concept's subconcepts. More formally, we say that a feature F is *subconcept independent* of a concept C_k, if and only if

$$p(C_i|f, C_k) = p(C_i|C_k) \tag{5}$$

for all feature values f of F and for all subconcepts C_i of C_k. Intuitively, information about a feature that is subconcept independent of a concept does not affect the actor's belief about any of that concept's subconcepts. An equivalent criterion for

[3]The set of nodes $\{C_i\} \cup B^{C_i}(F)$ d-separate the node F from the set of nodes $B^S(F) \setminus B^{C_i}(F)$.

[4]See [1, 4] for a comparison of subconcept independence with "subset-independence" in similarity networks [5].

subconcept independence which can be easily derived using Bayes' rule is

$$p(f|C_i) = p(f|C_k) \tag{6}$$

for all feature values f of F and for all subconcepts C_i of C_k [1, 4]. Since C_i is *any* subconcept of C_k, it follows that if F is subconcept independent of C_k, then

$$p(f|C_i) = p(f|C_j) \tag{7}$$

for all feature values f of F and for any pair of subconcepts, C_i and C_j, of C_k. This means that the probability distributions for a subconcept independent feature are identical across all the subconcepts. Hence we only need to store the distribution explicitly at the highest convenient position in the pc-hierarchy. The subconcepts then "inherit" the feature's possible values and distribution from one of their superconcepts.

To illustrate the idea of inheritance, consider the fragment of the pc-net for "transient state," "tool exit" and "tool entry" shown in Figure 3. The feature "current" is subconcept independent of "transient state." We do not need to explicitly store the probability distributions for "current" in the pc-diagrams for "tool entry" and "tool exit." That is, we may "omit" these probability distributions (and hence the corresponding feature nodes) in their respective pc-diagrams. When needed, the probability values are filled in by inheriting them from "transient state."

IV. Building Categorization Decision Models

Given a set of sensor information, an actor may construct and solve a categorization decision model to decide on the best course of action. Figure 5 shows an example of a categorization decision model that our automated machining actor T1000 might construct. The node "state of the machine" represents a conceptual cover which is a set of mutually exclusive and exhaustive concepts describing the current state of the machine. An example of a conceptual cover is the set {"tool chatter", "tool wear", “tool breakage”, “sensor failure”, “transient state”, “within variability limit” } as shown in Figure 6. Features whose values have been observed are indicated in Figure 5 by informational arcs from their respective nodes to the decision node. To simplify the figure, we have used black dots at the end of the informational arcs to indicate their continuation into the decision node.

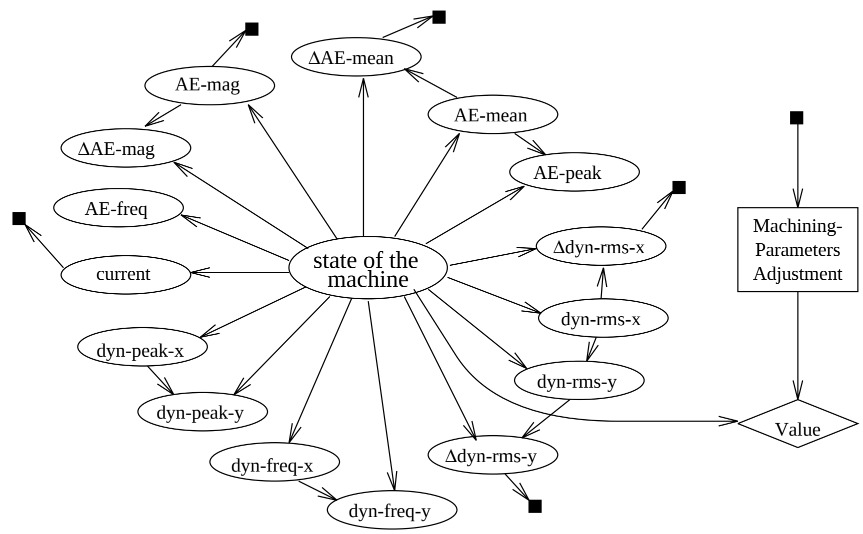

Figure 5: The categorization decision model

Given a conceptual cover z, let A_z^* denote the optimal action recommended by the corresponding categorization decision model. We define the *expected value of categorization* with respect to a conceptual cover z, denoted $\mathrm{EVC}(z)$, to be the expected value of following A_z^*. Let $C_c(z)$ be the computational cost associated with the corresponding categorization decision model. We define the *net expected value of categorization* with respect to conceptual cover z, denoted $\mathrm{NEVC}(z)$, to be $U(\mathrm{EVC}(z), C_c(z))$ where U is the overall utility function indicating the preference trade-off between EVC and computation cost. For the current implementation, we have assumed that the utility function U is of the additive form and EVC is in units

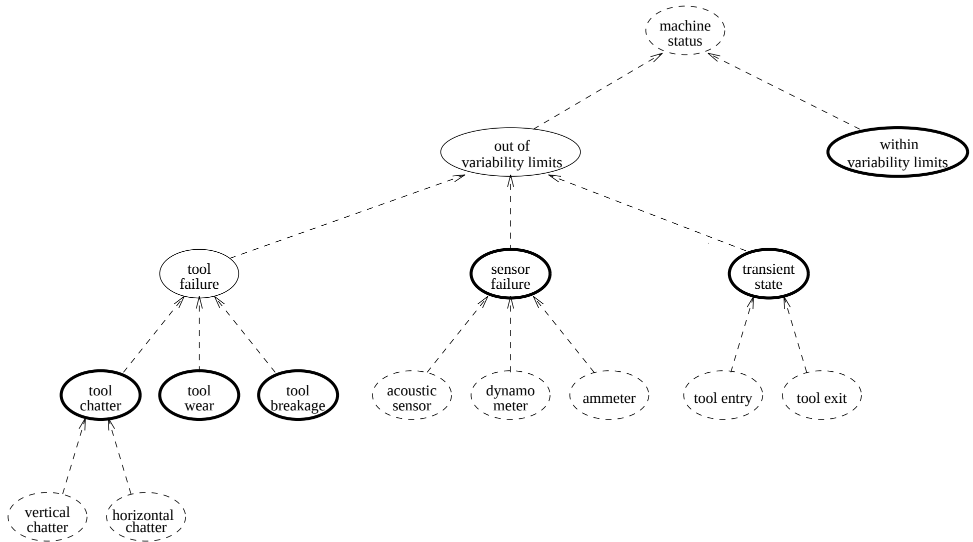

Figure 6: An example of a conceptual cover

of cost, i.e.,

$$\text{NEVC}(z) = \text{EVC}(z) - C_c(z). \tag{8}$$

In practice, the value of $C_c(z)$ is dependent on the level of abstraction z used to construct the categorization decision model, the topological structure of the decision diagram, and the algorithm for influence diagram evaluation. We have analyzed the computational cost of a number of categorization decision models with various graphical topologies [1] based on Shachter's influence diagram evaluation algorithm [6] and found that the computational cost generally increases rapidly with decrease in level of abstraction.

We now illustrate how a categorization decision model may be constructed from the pc-net for the automated machining problem. We shall assume the preferences are expressed by a utility function of the form $v(A_k, C_i)$ where A_k is the action taken and C_i the actual state of the machine. Possible actions include "reducing cutting speed", "reducing depth of cut", etc.

Suppose the sensors report information on "AE-mag," "AE-rms," "dyn-rms-x," "dyn-rms-y," and "rms-current," and T1000 decides to perform categorization at a level of abstraction corresponding to the conceptual cover {"tool chatter", "tool wear", "tool breakage", "sensor failure","transient state","within variability limits" }. We combine the respective pc-diagrams for these six concepts in the conceptual cover to construct a *categorization probabilistic influence diagram* (pid) as shown in Figure 7. The graphical structure of the combined categorization influence diagram is obtained by performing graphical union of the individual pc-diagrams while treating each central concept node as being the same node in each of the individual pc-diagrams. Notice that the concept node in the constructed diagram is now a probabilistic variable ($\mathcal{C}$) ranging over the six concepts used in its construction. The conditional probabilities for each of the feature nodes in the constructed diagram are obtained by copying over their respective values in the individual pc-diagrams.

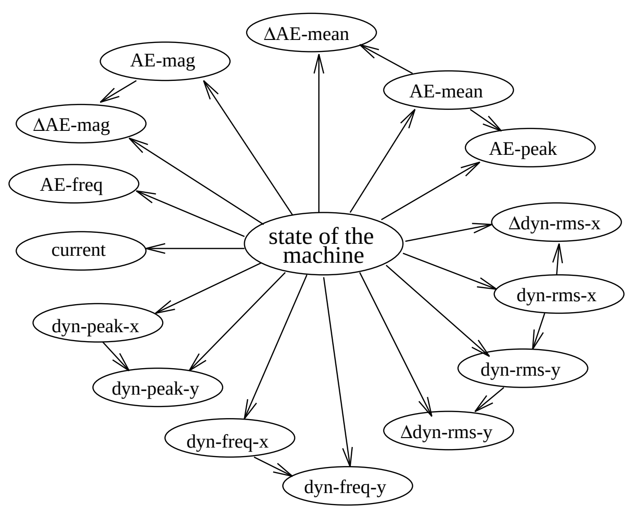

Figure 7: A categorization probabilistic influence diagram

The categorization decision model shown in Figure 5 is obtained from the categorization pid by adding the decision and value nodes to reflect the preferences of the actor, and then by adding informational arcs from the observed feature to the decision node are added. The completed categorization decision model can now be solved using existing methods (e.g., [6]). We formalize the categorization pid construction procedure as follows:

Construction 1 *A* categorization pid $\mathcal{G}$ *for a domain given conceptual cover* z *may be constructed from a pc-net* $\mathcal{N}$ *for the domain as follows:*

1. *Retrieve from the pc-net, the pc-diagrams for each concept in the conceptual cover. If a feature probability distribution is not explicitly stored in a pc-diagram, it inherits the distribution from the most specific super-concept which has the values in its pc-diagram.*

2. *By considering the category node in each of the pc-diagram to be the same node, form the graphical union of all the pc-diagrams producing the underlying graph for* $\mathcal{G}$*. Denote by* $\mathcal{C}$ *the central category node.*

3. *The marginal probability distribution* $p(\mathcal{C})$ *for the central node* $\mathcal{C}$ *is obtained from the subsumption probabilities for each concept given the root concept of the pc-net.*

4. *Fill in the conditional probability distribution for each feature node* F *in* $\mathcal{G}$ *by merging all the individual conditional probability distributions originally in the pc-diagrams as follows:*

$$p(F|C_i, B^{\mathcal{G}}(F)) = p(F|C_i, B^{C_i}(F)) \tag{9}$$

We say that the pc-net $\mathcal{N}$ constructs *the categorization pid* $\mathcal{G}$ *at level of abstraction* z *and use the notation* $\mathcal{N} \mapsto_z \mathcal{G}$*.*

We now formalize the construction of a categorization decision model from a categorization pid as follows:

Construction 2 *A categorization decision model* $\mathcal{D}$ *may be constructed from a categorization pid* $\mathcal{G}$ *as follows:*

1. *Construct a set of alternative actions including in it the most appropriate actions or courses of action that should be taken with respect to each concept in the conceptual cover.*[5]

2. *Add the action node to the diagram and include informational arcs from features which had been observed to the action node.*

3. *Obtain the preference information $v(A_k, C_i)$ for each possible action A_k and each possible concept C_i in the conceptual cover used to construct $\mathcal{G}$.*

4. *Add the value node to the diagram and add arcs from the action and category nodes to the value node.*

In any knowledge-based probabilistic model construction, it is important that probabilistic relations such as independence assertions, which are expressed in the knowledge base are preserved in the constructed model. This preservations of probabilistic relations can be characterized by the preservation of the joint distributions for the variables involved in the construction [5]. Given a set of concepts whose graphically consistent pc-diagrams it can be shown that the joint distribution for the variables $\mathcal{C}$ whose values range over the concepts and all the features in the pc-diagrams, is uniquely defined. Furthermore, this joint distribution is preserved across the categorization pid construction procedure [1].

V. Categorization Decision Model Improvement

In principle, an actor should consider all possible categorization decision models and pick the one with the maximum net expected utility of categorization. In practice,

[5]We assume that a general knowledge base exists which associates actions and preference information with each of the categories in the pc-net. A detailed examination of methods for constructing decision options and value models is a difficult problem that we are now addressing but is beyond the scope of this paper.

such an approach would be highly expensive in terms of computational cost. We propose an incremental approach to utility-based categorization by iteratively changing the level of abstraction within the conceptual cover used for model construction. There are numerous advantages in adopting an incremental approach. First, an incremental approach has approximately the so called *flexible*[7] or *any time* property [8, 9]. These are computational approaches which can be terminated prematurely during its computation and a partial result returned. In contrast, a computational approach which does not have the any time property would return nothing useful if it is terminated prematurely. It has been argued that flexible or anytime approaches are more suited for operation under conditions of great uncertainty in resource constraints [10]. Second, by applying the principles of decision-theoretic control of reasoning [10, 11, 12, 13], the problem of resource limitation for model construction and refinement can be computationally managed [14, 15]. Finally, since the categorization model is being modified iteratively, we can include any newly acquired information into the model at the end of each iteration. In contrast, a global solution approach would not be able to do so until it completes its computation.

Two possible operations may be performed on a conceptual cover to either increase or decrease the level of specificity of the concepts concerned. These operations are as follows:

1. *Conceptual specialization* whereby a concept in the conceptual cover is modified by replacing it with the set of its most general subsumees (i.e., subconcepts).

2. *Conceptual generalization* whereby a subset of concepts in the conceptual cover are replaced with their most specific subsumer (i.e., superconcept).

Figure 8 illustrates the conceptual specialization and generalization operations. Given the conceptual cover comprising the concepts "tool chatter," "tool wear," "tool breakage," "sensor failure," "transient state" and "within variability limits," the concept "transient state" may be specialized by replacing it with the set of its

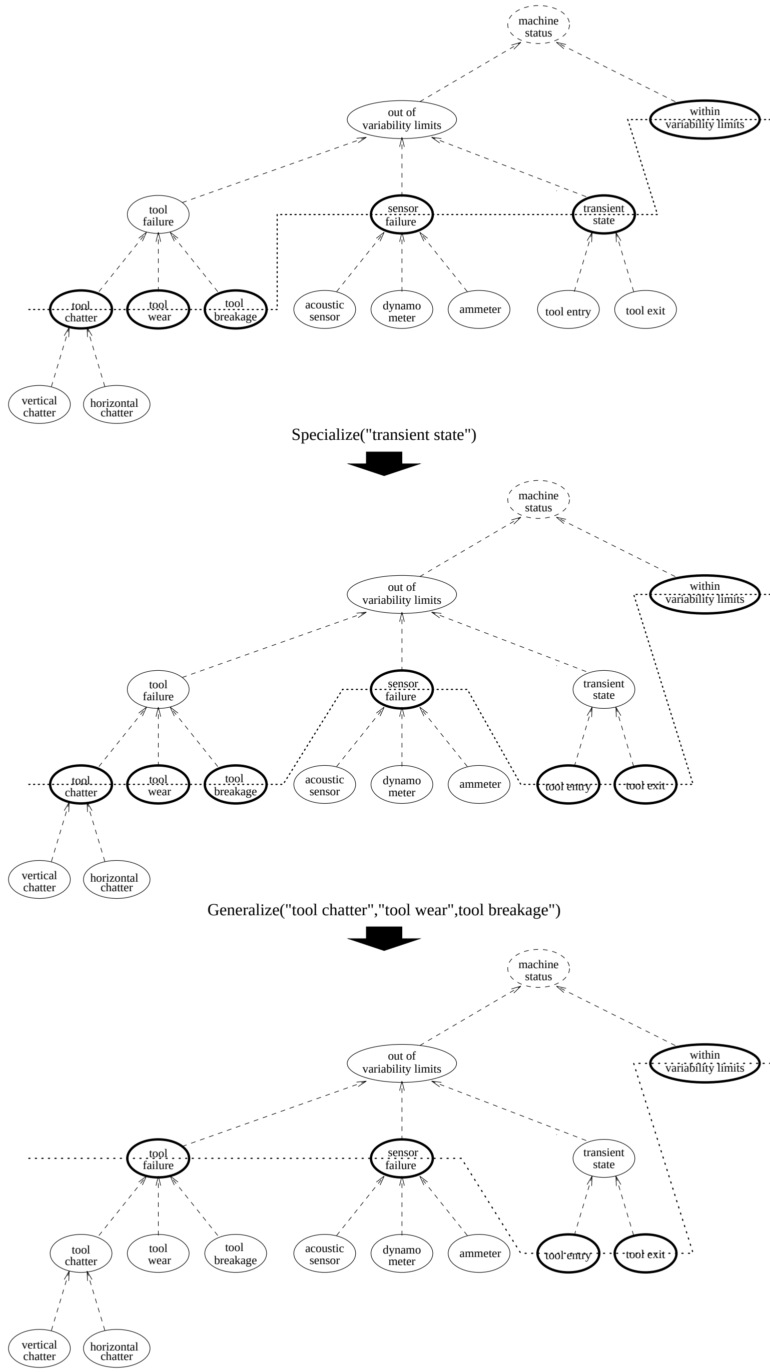

Figure 8: Conceptual cover specialization and generalization operations

most general subsumees, namely “tool entry” and “tool exit.” This results in a new conceptual cover corresponding to the set of concepts comprising “tool chatter,” “tool wear,” “tool breakage,” “sensor failure,” “tool entry,” “tool exit” and “within variability limits.” Similarly, in Figure 8, given the conceptual cover comprising the concepts the concepts “tool chatter,” “tool wear,” “tool breakage,” “sensor failure,” “transient state” and “within variability limits,” the three concepts “tool chatter,” “tool wear” and “tool breakage” may be generalized and replaced by their most specific subsumer “tool failure.” A new conceptual cover comprising the concepts “tool failure,” “sensor failure,” “transient state” and “within variability limits,” may be formed in this way.

Given a conceptual cover z_o, we will denote a model improvement operation by s_i and $z_i = s_i(z_o)$ the resulting conceptual cover. As a convention, we will denote the null operation by s_o. We can also perform a sequence of n model improvement operations $s_1, s_2, \ldots, s_n$ such that the resulting conceptual cover is $z_{n+1} = s_n(s_{n-1}(\cdots s_1(z_o)))$.

VI. Control of Model Improvement

We develop measures that aid a categorical reasoner in estimating the values obtained by varying the level of categorical abstraction in a categorization decision model. We define the *expected value of model improvement* (EVMI) for operation s_i on conceptual cover z_o to be

$$\mathrm{EVMI}(s_i) = \mathrm{EVC}(s_i(z_o)) - \mathrm{EVC}(z_o) \tag{10}$$

where $s_i(z_o) = z_i$ denotes the resulting conceptual cover. In practice, EVMI values can be estimated for categorization decision models with certain types of topology, e.g., when all the features are conditionally independent of each other given the central concept [1].

EVMI values offer plausible guidance in controlling categorical reasoning. We have developed control techniques for making decisions about categorization decision model improvement. To consider the use of EVMI measures, we must balance the expected benefits of model improvement in terms of the expected value of the recommended action with the corresponding change in computation cost. We define the *net expected value of model improvement* (NEVMI) as the difference between the EVMI and change in model computation cost and the cost of making the improvement. That is,

$$\mathrm{NEVMI}(s_i) = \mathrm{EVMI}(s_i) - \Delta C_c(s_i) - C_g(s_i) \tag{11}$$

where $\mathrm{EVMI}(s_i)$ is the expected value of model improvement for operation s_i, $\Delta C_c(s_i) = C_c(z_i) - C_c(z_o)$ is the change in computation cost of the model due to the operation s_i, and $C_g(s_i)$ is the cost of performing the model improvement operation.

The identification of a theoretically optimal sequence of operations requires a combinatorial search through all possibilities. In practice we can approximate the process by employing a myopic or *greedy* NEVMI control procedure [10]. At each iteration, we seek to identify the *single-step* model improvement operation with the greatest NEVMI, i.e.,

$$\arg\max_{s_i} \mathrm{NEVMI}(s_i). \tag{12}$$

We iteratively repeat this greedy analysis and halt model improvement when all operations have $\mathrm{NEVMI}(s_i) \leq 0$. Figure 9 shows the fragment of the graph of possible model improvement steps.

The myopic approach works fine if all NEVMI values diminish monotonically, but it can overlook positive NEVMI values that lie more than one step ahead of the current state. We can relax the myopia of the NEVMI analysis by allowing varying amounts of lookahead. For example, we can consider the NEVMI of two or more model improvement steps. Such lookahead can be invoked when single-step analysis yields negative NEVMI values for all operations. We can also generalize the analysis to a general n-step look ahead procedure. By careful experimentation, one may learn about the characteristics of the utility changes that accrue from stepwise model improvements and adopt a plausible value for lookahead.

The analysis described above is based on an *urgency* model of computational cost

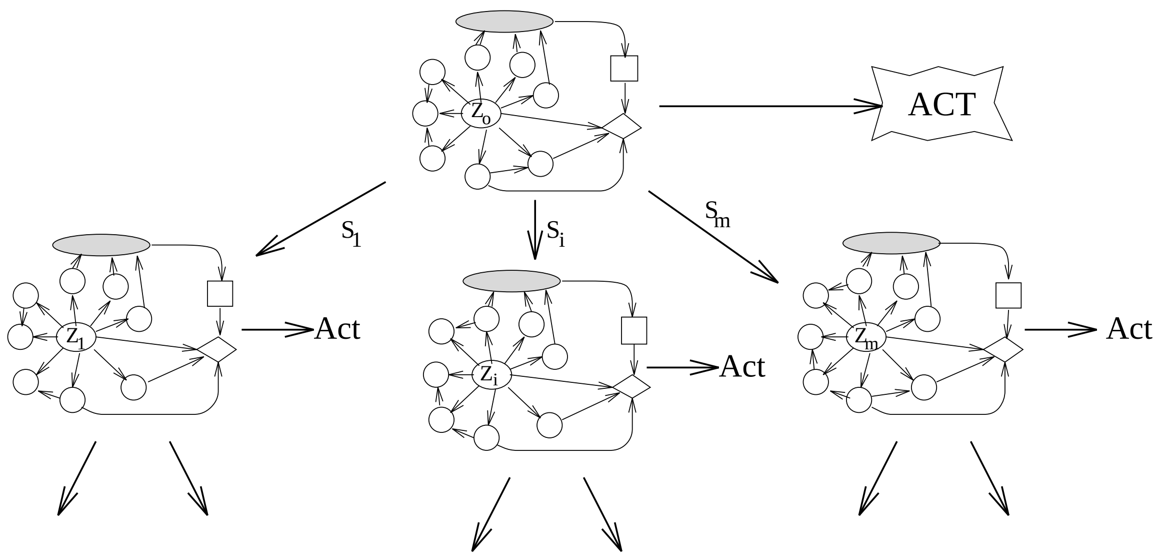

Figure 9: Greedy control of model improvement

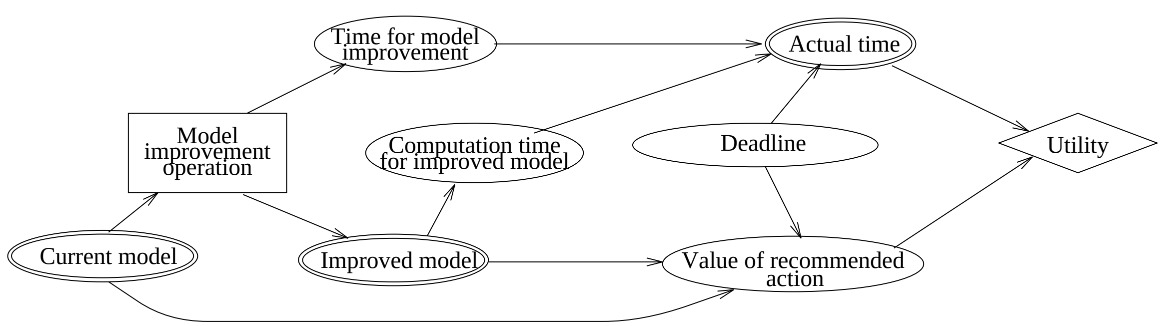

Figure 10: Model improvement under an uncertain deadline

[10] where utility decreases monotonically with the amount of computational cost. For the case where there is an uncertain deadline, we present the model improvement control problem using the influence diagram shown in Figure 10. As before, let z_o represent the current level of abstraction. Let s_i $(i = 1, \ldots, p)$ be the set of possible model improvement operations applicable on z_o such that $s_i(z_o) = z_i$. Let s_o represent the null operation, i.e., $s_o(z_o) = z_o$. Let the uncertain deadline be denoted by t_d. Let $t_c(z_i)$ represent the uncertain model computational time requirement for a model corresponding to z_i. Let $t_g(s_i)$ be the uncertain model improvement time associated with operation s_i. Depending on whether the total time $t_c(z_i) + t_g(s_i)$ exceeds the deadline t_d or not, the actual time utilized is $T_t(s_i) = \min(t_c(z_i) + t_g(s_i), t_d)$. Let $\mathrm{EVC}(z_i)$ denotes the value of the recommended action given that operation s_i is applied.

Given a model improvement operation s_i, we consider two possibilities. In the first case, when $t_c(z_i) + t_g(s_i) \leq t_d$, there is enough time to complete the evaluation. The overall utility of the outcome is then $U(\mathrm{EVC}(z_i), C_a(t_c(z_i) + t_g(s_i)))$ were C_a is a function converting computation time to cost.[6] In the second case, when $t_c(z_i) + C_g(s_i) > t_d$, there is not enough time to complete the evaluation. Without model improvement, the actor would not be able to follow any newly recommended action and would continue to act without it. The utility of the outcome for this second case is $U(\mathrm{EVC}(z_o), C_a(t_d))$. The net expected value of model improvement for s_i is therefore

$$
\begin{aligned}
\mathrm{NEVMI}(s_i) &= p(t_c(z_i) + t_g(s_i) \leq t_d)[\mathrm{EVC}(z_i) - C_a(t_c(z_i) + t_g(s_i))] + \\
&\quad p(t_c(z_i) + t_g(s_i) > t_d)[\mathrm{EVC}(z_o) - C_a(t_d)] - \mathrm{EVC}(z_o)
\end{aligned}
$$

We can simplify the NEVMI to

$$
\mathrm{NEVMI}(s_i) = p(t_c(z_i) + t_g(s_i) \leq t_d)[\mathrm{EVMI}(s_i) - C_a(t_c(z_i) + t_g(s_i))] -
$$

[6] We are assuming that computation time is the only factor affecting cost.

$$p(t_c(z_i) + t_g(s_i) > t_d)C_a(t_d) \tag{13}$$

VII. An Application in Automated Machining

To demonstrate the benefits of our system over one which does not allow for flexibility of abstraction in conceptual representation, we will illustrate how our system might possibly guide the behavior of our automated machine (T1000) under different situations and context.

Suppose T1000 had just been set up for a new job with new set of cutting tools and its sensors were checked recently. We can assessed low prior probabilities to the concept "out of variability limits" and two of its subconcepts "tool failure" and "sensor failure" in the pc-net. Sensor readings were obtained and the meta-level decision model indicated that the optimal conceptual cover for the current situation was "out of variability limits", "within variability limits". The recommended action was "do nothing".

A little while later, the sensors picked up an increase in acoustic emission ("AE-mag") and an increase in cutting force variation in the x direction ("Δdyn-rms-x"). Based on these information, the system indicated that specializing the conceptual cover to {"tool wear", "tool chatter", "tool breakage", "sensor failure", "transient state", "within variability limits"} would yield the highest NEVMI with the recommended action "reduce depth of cut to 0.5mm". After the recommended action was carried out, the sensor readings began to change and the system returned to the conceptual cover {"out of variability limits", "within variability limits" } as before.

As the machining proceeded, we updated the pc-net to reflect a higher possibility of tool wear based on the amount of usage on the tool. An hour later, again based on sensor readings, the system changed the conceptual cover in the model to {"tool failure", "sensor failure", "transient state", "within variability limits"}. This was due to the fact that the tool had been used sufficiently long enough to warrant a change in level abstraction in the model. The system also recommended a lowering of the cutting tool feed rate to one-half of the current value.

We observed that by allowing for flexible abstraction of conceptual representation, our system was able to guide an actor to respond with the best course of action depending on the situation and context. A conventional categorization system, on the other hand, would always perform its reasoning based on a fixed conceptual cover, perhaps at the most detailed level. It would not be able to flexibly adjust its level of conceptual representation and reasoning according to the situation and context. For example, when caught in a critical situation such as, for example, when T1000 broke its tool and any prolonged delay in changing the tool would have damaged the workpiece, our system can respond appropriate by assigning a high cost to computation time and work on a model containing just sufficient information to see it through the situation. On the other hand, a system with fixed conceptual representation or one which always seek the most comprehensive conceptual description would not be able to respond adaptively to the criticality of the situation.

VIII. Discussion

We have described a dynamic model construction process within an overall decision-theoretic framework for categorization. Our approach is strongly motivated by an intelligent actor's need to manage the tradeoff between the potential benefits of using more elaborate concepts and the computational costs that may be incurred when using them to reason about action. The hierarchical pc-net in our approach to categorical reasoning incorporates inheritance mechanisms to economize on the information stored at any given level of abstraction in a hierarchy. Rather than rely upon this hierarchical, inheritance-based approach, it is possible to store all knowledge at the most specific conceptual cover (i.e., the lowest hierarchical level or "leaves" in the pc-net). From the decision model corresponding to this conceptual cover, one could derive all the other more abstract decision models. However, our use of hierarchical representation and inheritance mechanisms seems preferable for modeling the problem-solving of resource-constrained actors. By enabling the detection and use of subconcept independence relations, our approach provides a form of "cognitive economy" that is likely to greatly simplify the elicitation process (cf. [5]). The pc-net representation might also be viewed as a form of generalization of Heckerman's two level hierarchy which is based on the notion of subset independence.

Our research on utility-based categorization grew out of our interest in more traditional approaches to knowledge representation and reasoning about concepts found in the AI and cognitive science literature (cf. [1] for a thorough review). The notion of concept-subconcept hierarchies is a central theme in that literature, although a great deal of attention is given to methods to overcome the limitations of strict hierarchies when deterministic relations are presumed between concepts and subconcepts or between concepts and features. This includes the use of multiple inheritance representations in conjunction with defaults, overrides, and other "non-monotonic" inference devices. Our approach also presumes hierarchical organization, but allows for probabilistic concept-subconcept and concept-feature relations. This probabilistic approach overcomes some of the limitations of deterministic approaches regarding concept-feature relations. Our use of hierarchical concept-subconcept relations may benefit from being generalized to the use of probabilistic, multiple-inheritance relations.

Our focus on categorical reasoning as an element of reasoning about action is a particularly distinctive aspect of this research. By viewing choice of representation in terms of its ultimate impact on effectiveness (i.e., utility) of action, we provide the basis for a truly normative theory of categorization which prescribes how a resource-constrained actor, given some sensory information, should most effectively exploit its beliefs about concepts and their features in order to maximize the effectiveness of that actor's impact on the world. One challenging extension to this work would be to generalize the model construction methods describe herein for the case of "heterarchical" organization of concepts (i.e, probabilistic relations to multiple parents of a given subconcept). This extension poses some formidable mathematical challenges that we shall address in the near future.

Other promising extensions of our work include development of methods to automatically generate an abstraction hierarchy from lowest-level concepts or the removal of intermediate level concepts that are irrelevant to the problem or task at hand.[7] One promising approach to utility-based clustering involves starting from the most specific conceptual cover, recursively merging two or more concepts in the cover whose generalization has the highest EVMI value, and stopping when the next generalization produces a single (root) concept.

The idea of using expected value of decision modeling was first addressed in [17], as well as [18]. Categorical specialization and generalization operations on general probabilistic models have also been studied in [19] where they investigated the problem of dynamically refining and coarsening of state variables in probabilistic influence diagrams and specified a set of constraints that must be satisfied to ensure that the coarsening and weakening operations do not affect variables that are not involved. In particular, the joint distribution of the Markov blanket excluding the state variable itself must be preserved. However, the value and cost of performing such operations were not addressed. The idea of attention focusing in decision making was also studied in [20]. They investigated the value of "extending the conversation" by identifying additional conditioning variables that might be added to existing decision models.

In related work, we have also investigated some general techniques for reasoning about the value of refining general decision models [14, 15]. These techniques for refining general decision models can be incorporated into automated reasoning systems or computer-based decision modeling and evaluation systems [21, 22]. These techniques do not make any presumption of explicit availability or representation of knowledge. By applying similar principles for the decision-theoretic control of reasoning, we can provide guidance on when it is best for an actor to cease further refinement of a decision model and take immediate action in the world.

[7] Several methods for generating concepts based on utility-based clustering are described in [16].

IX. Conclusions

The methods for dynamic model construction presented here fall within a novel framework for utility-based categorization as a type of deliberation in support of reasoning about action. Elsewhere, the first author and his colleagues discuss other important aspects of this research (e.g., [1, 4, 14, 15]).

The model-construction approach presented here is motivated by recognition of a resource-constrained actor's need to manage the difficult tradeoff between the benefits to be gained from representing its problems by using more detailed categories and the possibly unacceptable computational costs that use of such detailed descriptions might entail. In order to address this problem we have also introduced a single-inheritance-based probabilistic network representation scheme.

We believe that our approach, particularly, the use of decision-theoretic control of model modification is an important new direction for knowledge-based model construction applications. It provides a basis for a promising approach to balancing the computational costs and actional benefits of the model construction process. We have illustrated the application of our concept-representation and model-construction methods with an example from manufacturing automation and showed how our system can guide an automated machining operation through different situation and context. We also believe that our approach offers equal value in addressing similar issues in many other application domains, most notably for the domain of medical diagnosis. The first author is currently exploring the use of the approach to categorize personnel records in a large organization in Singapore.

Acknowledgement

The authors would like to thank the editors and the anonymous refees for their invaluable suggestions which had help to significantly improve the paper.

References

[1] K. L. Poh, *Utility-Based Categorization*, PhD thesis, Department of Engineering-Economic Systems, Stanford University, 1993.

[2] E. E. Smith and D. L. Medin, *Categories and Concepts*, Harvard University Press, Cambridge, Massachusetts, 1981.

[3] A. M. Agogino and K. Ramamurthi, "Real time influence diagrams for monitoring and controlling mechanical systems", in *Influence Diagrams, Belief Nets and Decision Analysis*, R. M. Oliver and J. Q. Smith, Eds., pp. 199–228. John Wiley, 1990.

[4] K. L. Poh and M. R. Fehling, "Probabilistic conceptual network: A belief representation for utility-based categorization", in *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, 1993, pp. 166–173.

[5] D. E. Heckerman, *Probabilistic Similarity Networks*, MIT Press, Cambridge, Massachusetts, 1991.

[6] R. D. Shachter, "Evaluating influence diagrams", *Operations Research*, vol. 34, no. 6, pp. 871–882, Nov–Dec 1986.

[7] E. J. Horvitz, "Reasoning under varying and uncertain resource constraints", in *AAAI-88, Proceedings of the Seventh National Conference on Artificial Intelligence*, 1988, pp. 111–116.

[8] T. Dean and M. Boddy, "An analysis of time-dependent planning", in *AAAI-88, Proceedings of the Seventh National Conference on Artificial Intelligence*, 1988, pp. 49–54.

[9] T. Dean, "Decision-theoretic control of inference for time-critical applications", *International Journal of Intelligent Systems*, vol. 6, no. 4, pp. 417–441, 1991.

[10] E. J. Horvitz, *Computation and action under bounded resources*, PhD thesis, Departments of Computer Science and Medicine, Stanford University, 1990.

[11] M. R. Fehling and J. S. Breese, "A computational model for decision-theoretic control of problem solving under uncertainty", Technical Memo 837-88-5, Rockwell International Science Center, Palo Alto Laboratory, Palo Alto, California, 1987.

[12] E. J. Horvitz, "Reasoning about beliefs and actions under computational resource constraints", in *Proceedings of the Third Workshop on Uncertainty in Artificial Intelligence*, 1987, pp. 429–439.

[13] S. Russell and E. Wefald, "Principles of metareasoning", *Artificial Intelligence*, vol. 49, pp. 361–395, 1991.

[14] K. L. Poh and E. J. Horvitz, "Probing the value of decision-model refinement", Technical Report 85, Palo Alto Laboratory, Rockwell International Science Center, Palo Alto, California, Dec. 1992.

[15] K. L. Poh and E. J. Horvitz, "Reasoning about the value of decision-model refinement: Methods and application", in *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, 1993, pp. 174–182.

[16] E. J. Horvtiz and A. C. Klein, "Utility-based abstraction for categorization", in *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, 1993, pp. 128–135.

[17] S. R. Watson and R. V. Brown, "The valuation of decision analysis", *Journal of the Royal Statistical Society*, vol. 141, no. Part 1, pp. 69–78, 1978.

[18] R. C. Nickerson and D. W. Boyd, "The use and value of models in decision analysis", *Operations Research*, vol. 28, no. 1, pp. 139–155, 1980.

[19] K. Chang and R. Fung, "Refinement and coarsening of Bayesian networks", in *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990, pp. 475–482.

[20] David E. Heckerman and Holly Jimison, "A perspective on confidence and its use in focusing attention during knowledge acquistion", in *Proceedings of the Third Workshop on Uncertainty in Artificial Intelligence*, 1987, pp. 123–131.

[21] E. J. Horvitz, J. S. Breese, and M. Henrion, "Decision theory in expert systems and artificial intelligence", *International Journal of Approximate Reasoning*, vol. 2, pp. 247–302, 1988.

[22] S. Holtzman, *Intelligent Decision Systems*, Addison-Wesley, 1989.