Steps Toward Automating Decision Making

COINS Technical Report 87-82

Adele E. Howe and **Paul R. Cohen**

Experimental Knowledge Systems Laboratory
Department of Computer and Information Science
University of Massachusetts, Amherst, MA
August, 1987

Abstract

Standard models of decision making provide normative methods for analyzing quantitative assessments of decision situations. These models are inadequate for most problems in artificial intelligence because they do not specify how to acquire assessments, and they depend on a complete, combinatorial model that requires too many data to assess. This paper presents a brief overview of some of these models and proposes an alternative methodology, ***constructive decision making***, that addresses some shortcomings of normative models in knowledge based paradigms. An implementation of the constructive decision making approach is described.

This research is funded by DARPA/RADC Contract F30602-85-C-0014, by National Science Foundation Grant IST 8409623, and by ONR on University Research Initiative Grant N0001486K0764.

1 Introduction

Making a decision is the process of accumulating and integrating evidence, selecting an alternative, and perhaps acting on that selection. In formal decision-making, the first step is typically performed by a decision analyst and a client in a relationship similar to that of a knowledge engineer and an expert. Together, they build a complete *model* of all the factors relevant to the outcomes of the alternatives. Then, they assess probabilities and utilities for these outcomes given the factors. Once the model is constructed, simple algorithms can select an alternative that is optimal with respect to the model. If the model is to faithfully represent the real world for the purpose of selecting the best alternative, then it must capture the combinatorial space of all outcomes of all alternatives conditioned on all factors. By AI standards, searching this space is relatively inexpensive, but the cost of constructing it can be enormous. This is for two related reasons. First, it is done by people, not computers, and involves the same kind of ponderous, error-ridden interviewing that characterizes the knowledge acquisition bottleneck in knowledge engineering. Second, by keeping separate the construction and selection processes, the constraints produced as part of selection cannot be used during construction to reduce the search space that must be examined, and so the model may include information that is not relevant to the selection process.

This paper presents an alternative approach called ***constructive decision making*** that merges the construction and selection processes. It iteratively asks whether more information could increase confidence in a decision and, if so, decides what information is needed. It views decision-making as a transition through ***decision states***, each of which represents a decision supported by successively more evidence. Thus, the algorithm can offer a decision at any time, with the proviso that more evidence might result in a better decision. Currently, the constructive decision making approach supports two-alternative, multiattribute decisions; a multi-alternative version of the approach is under development.

Constructive decision making underlies a decision support system that incrementally identifies the factors that influence a decision, and moves from states where alternatives have weak support to states in which choices are more clear. This system, called CDM, emphasizes the process of acquiring and structuring just the information required for a decision. It characterizes the current state of the decision with respect to strength of support for each alternative. If no alternative is clearly superior, it seeks information about factors of the decision that can discriminate the alternatives. It never requests information that it does not need to develop the decision. The process of acquiring information about attributes continues until a decision is forced or a clear choice emerges.

Since constructive decision making merges the processes of constructing a decision and selecting an alternative, it is ideally suited to AI programs that must construct decisions for themselves. This situation arises in domains where the relevance of factors cannot be determined until run-time; that is, domains where decision analysts cannot construct a combinatorial model of a decision ahead of time. For example, programs with dynamic control structures must construct dozens or hundreds of control decisions based on factors whose relevance changes in the course of problem-solving [1,3,6]. Moreover, in real-time control problems, the constructive decision making approach can offer satisficing decisions that are the best possible given the time available to collect and process evidence.

In overview, this paper first discusses our motivations for designing the constructive decision making approach, focusing on traditional decision science and relevant literature on human judgment. Next, we present the theory of constructive decision making, followed by a discussion and illustration of the CDM decision support system. We conclude with an assessment of future work in the area.

2 Decision-theoretic methods and human judgment

Decision theory was developed to produce optimal decisions in difficult or complex problems. It has become the theoretical basis for most other work in decision making because it provides a consistent, rigorous mathematical representation of decisions under uncertainty [14], and offers a definition of optimality that depends on the available knowledge [13].

2.1 Decision analysis

Decision analysis is a decision-making methodology based on decision theory [20]. A central concept in decision analysis is the model of a decision, usually represented as a decision tree, which includes the alternatives, their outcomes, and all the chance factors that affect the probabilities of the outcomes. Models provide the ability to assess the utility of evidence, and conversely, to assess the cost of uncertainty. They encode subjective information about risk aversion and probability, and they justify decisions [8,18].

However, constructing models of decisions is a painstaking, time-consuming process that usually requires the assistance of a trained decision analyst. Models require two kinds of information, conditional probabilities and utilities, about the outcomes of alternatives. Probability assessments are never clear-cut and are often difficult to obtain [18]. Utilities must be expressed on a single scale, such as monetary worth, which forces people to place monetary values on distinctly non-monetary outcomes such as environmental damage or an education [9]. Multiattribute decision theory addresses this point, but at the cost of increased complexity [22]. Moreover, a complete model must be specified *before* a decision can be analyzed; and because models are combinatorial (all outcomes of all alternatives are conditioned on all factors), a decision analysis swiftly becomes a "bushy mess" for problems involving more than a few alternatives[18]. Skilled decision analysts can reduce this complexity by narrowing the number of distinct alternatives considered for each action and eliminating "impossible" conditional combinations of alternatives, chance factors, and outcomes (e.g. [7]), but most agree that decisions involving more than 10 chance factors are extremely difficult to model.

2.2 Decision-support systems and AI decision-making systems

Because it is difficult to construct decision models, decision support systems have been designed to assist people in solving unstructured problems by providing specially tailored models and relevant historical data [5]. Most decision support is designed for specific areas such as corporate planning, portfolio management, and marketing, and uses mathematical models of decision making expertise [23]. These systems assist decision making by providing access to supportive data and simulations of possible effects of a decision selection in tightly constrained domains. Consequently, model-based decision support reduces the informational demands on the user, but does not address how to build the models or make the selection. A few domain-independent systems have been developed to assist in modeling decisions. ARIADNE, Alternative Ranking Interactive Aid based on DomiNance structural information Elicitation, addresses the problem of eliciting from a user a dominance structure for selecting from multiple criteria alternatives [19]. Leal and Pearl describe a program that interacts with the user to construct decision trees [12]. GODDESS, a GOal-Directed DEcision Structuring System, produces a hierarchical goal representation of decision alternatives by selectively focusing the user's attention on the most crucial issues [17].

We began the research reported here by asking whether decision-making in AI programs must necessarily be based on decision theory. If not, then an approach might be developed that enables AI programs to build their own decision models and evaluate them without the intervention of a decision analyst. We have focused on questions about three aspects of decision-theoretic models: first, can AI programs evaluate alternatives on several criteria without getting involved with the complexity of multiattribute decision theory? second, is it necessary to build a complete model before evaluating alternatives?; and third, is the optimality criterion appropriate? The first question arises because decisions in many AI tasks involve many criteria that are not easily reduced to a single utility scale; for example, the decision to order a painful, dangerous test in medicine depends on its morbidity and mortality rates, psychological stress to the patient, the monetary cost of the test, the degree to which performing the test obliges the physician to order more serious tests, and so on. The last two questions are closely related; the traditional division between building a model and evaluating alternatives exists to ensure optimal decisions. By examining the utilities and probabilities of *all* relevant outcomes of alternatives, one can guarantee selecting the optimal outcome, but this means that no alternative can be selected until the entire model is constructed.

Many decisions must be made in dynamic situations by AI programs without human intervention. In these cases, a program must define the decision by itself. Furthermore, if the program is required to work in real time, or with unknown deadlines, it must avoid combinatorial decision models and, most importantly, it must evolve decisions as evidence becomes available instead of demanding a complete model in advance of making a decision. That is, algorithms for programs that construct their own decisions must:

1. not consume resources building a complete combinatorial model
2. not hold up processing while waiting for evidence
3. provide the 'best' decision at any point in processing, although supporting this capability will sacrifice the goal of optimality
4. work even when the alternatives and attributes are not known or specified a priori, but emerge in the course of decision making.

These capabilities are not supported by traditional decision science. AI systems must respond to dynamic environments in which precise information is not always available and decisions are more or less certain depending on the available evidence, and, thus, to the time allowed to collect and assimilate evidence. In short, we require graceful degradation of the decision-making performance of AI systems when time and evidence are in short supply.

2.3 Human judgment and decision making

Most human decisions are taken without the support of decision analysis, although rough, qualitative versions of decision-analytic concepts underlie everyday decisions. For example, when deciding to purchase one automobile instead of another we consider factors such as reliability, comfort, handling, and so on. Rough probabilities, such as the chance of major repairs within the warrantee period, may be considered. An inexpensive car with a moderate probability of needing minor repairs may be preferred to a very expensive one with a lower probability of failure. This is essentially decision-analytic reasoning. But this "naive decision analysis" differs from the real thing in two respects. First, it relies on assessments that are quite likely to overestimate or underestimate true probabilities. Second, it is much more dynamic than decision analysis, intermixing the tasks of constructing and evaluating decision models, often revising alternatives and attributes, and relying on simple heuristics to keep track of the relative merit of alternatives. The first point, amply illustrated by the literature on human judgment under uncertainty (e.g., [11,16]) suggests that AI decision making programs should use true probabilities or be robust against inaccuracies in subjective probabilities. This is the subject of a number of papers in Artificial Intelligence, (e.g., [4,7]). The second point, however, suggests alternatives to the traditional division between constructing and evaluating decision models: Perhaps human heuristics for dynamically constructing decisions can be used in AI programs.

Protocol studies of human decision makers show that they do not first list all alternatives, outcomes, chance factors, conditional probabilities, and criteria for evaluating utilities of outcomes; and then build a decision tree. On the contrary, the structure of decisions apparently emerges dynamically. In one study, analysis of four senior auditors working on an audit case showed that in the course of the decision process, the operations of structuring, search and analysis were intermixed, and that evaluative criteria emerged as the decision progressed [2]. Similar results were obtained by Ola Svenson in studies of people selecting a house. He found that evaluative criteria varied across subjects and became more specific and detailed as the decision progressed. Additionally, new criteria tended to emerge in the course of making the decision [21]. To manage such dynamic structuring and revision, humans apparently use a few simple heuristics. For example, Payne, Braunstein and Carroll [15] found that subjects eliminate alternatives by deleting any whose attributes don't equal a criterion value, and then compare pairwise the remaining alternatives. Bettman and Jacoby found that subjects used a strategy called "choice by feedback processing" in which they alternated between considering alternatives with respect to attributes and attributes with respect to alternatives [21]. The most compelling aspects of these studies are that people successfully use these heuristics, often tailored to pairwise comparisons, to reduce the computational requirements of complex decision tasks; and that decision making is a constructive process that selectively includes, analyzes and discards information as it becomes more or less relevant to the decision. The design of the constructive decision making approach, described in the next section, has been influenced by these observations.

3 Constructive Decision Making

The development of constructive decision making (CDM) was motivated by the need for intelligent programs to define and evaluate decisions autonomously. It differs from traditional decision analysis in two respects: it has a simple mechanism for handling incomparable attributes, and it is constructive. In CDM, attributes such as cost and reliability are not collapsed to a single utility scale, although they may be grouped in support of the same alternative. Information about attributes is added to the evolving decision constructively, which means that the decision maker always has the choice of accepting a decision or adding information to increase confidence in the decision. The following sections describe the CDM approach to solving two alternative, multiattribute decision problems.

3.1 Decision Typology

The core of constructive decision making is the decision typology. The typology characterizes decision situations using domain-independent dimensions, guides the collection of support and provides the best possible decision given available evidence. The typology represents the current and potential future states of a decision and allows a program to

- select an alternative that is preferred given the available information
- actively reduce uncertainty in the selection of the alternative by collecting evidence to support it or refute it.

We developed the typology to model the process of making decisions based on incomparable attributes. We call problems of this type *apples and oranges problems.* When you compare apples and oranges in a grocery store you may find one fruit preferred on the basis of flavor and the other on the basis of quality. If you could combine the attributes to compare the alternatives on a single, composite attribute, then the choice is often clear. But if, as in this case, flavor and quality are not easily combined, then the choice between apples and oranges is problematic. The following description of the typology will refer back to this example.

3.1.1 Decision States

We begin the discussion with simple 2-alternative, 2-attribute problems typified by the apples and oranges problem. Decision alternatives are compared on their salient attributes.[1] A *decision state* in our model is a concise description of the current combination of salient attributes, including how well the alternatives are distinguished on the available attributes and how important those attributes are. We identified five *dimensions* that describe decision states.

The dimensions in the decision typology are abstracted from the decision situation and are used to reason about the state of the decision.

- significant difference with respect to attribute-1
- significant difference with respect to attribute-2
- conflict
- importance
- greater-than.

[1] Throughout this paper "attribute" is used loosely to refer to features of alternatives that are salient to the task of selecting the best alternative. This definition is vague enough to accommodate *outcomes, goals* or *characteristics.* The distinction will be refined in Section 3.1.4.

Significant difference with respect to an attribute indicates whether the values of the attribute for the two alternatives are distinct. Assuming that the values of all the other attributes for the two alternatives are equal, can a decision be made based only on *this* attribute? This dimension determines whether the difference between two alternatives is significant enough to support a decision. It effectively avoids the issue of exactly what value each alternative has or what distribution of values can be expected and how much difference is required to be significant. The formal definition[2] used in the typology is:

$$Sd[A_i] = \begin{cases} 1 & \text{if } A_i[p] \text{ and } A_i[q] \text{ are distinct} \\ 0 & \text{otherwise} \end{cases}$$

Otherwise indicates no significant difference or that we lack evidence to tell whether there is a significant difference.

A **conflict** exists when the two attributes support different alternatives, that is, we have conflicting evidence. Formally, this is described as:

$$C[A_i, A_j] = \begin{cases} 1 & \text{if } A_i[p] \tilde{>} A_i[q] \text{ and } A_j[p] \tilde{<} A_j[q] \text{ or} \\ & \text{if } A_i[p] \tilde{<} A_i[q] \text{ and } A_j[p] \tilde{>} A_j[q] \\ 0 & \text{otherwise} \end{cases}$$

Importance indicates whether one attribute is considered more influential than the other. Once again we have avoided the issue of *why* we believe this. It may be because the attribute itself is more important, independent of the values of the alternatives on the attribute, or that the values of the alternatives are so radically different on a particular attribute that it is more discriminating that the others.

$$I[A_i, A_j] = \begin{cases} 0 & \text{if importance}(A_i) = \text{importance}(A_j) \\ ? & \text{if relative importance is unknown} \\ 1 & \text{if importance}(A_i) > \text{importance}(A_j) \\ & \text{or importance}(A_i) < \text{importance}(A_j) \end{cases}$$

Greater_than is simply an extension of Importance to indicate *which* attribute is more important if, in fact, one is.

$$\tilde{>}[A_i, A_j] = \begin{cases} 0 & \text{or importance}(A_i) < \text{importance}(A_j) \\ 1 & \text{if importance}(A_i) > \text{importance}(A_j) \end{cases}$$

These dimensions can be illustrated in the context of the problem of selecting fruit: p is apples, q is oranges, A_i is *quality* and A_j is *flavor*. If the quality of apples is "good" and the quality of oranges is "poor", then $Sd[quality] = 1$ because good and poor are distinct values. Similarly, if one prefers the flavor of oranges to that of apples then $Sd[flavor] = 1$. Since apples have better quality but oranges taste better, $C[quality, flavor] = 1$. Finally, if quality is preferred to taste $I[quality, flavor] = 1$ and $\tilde{>}[quality, flavor] = 1$. These dimension values can be put together to form a vector descriptor of the state of the decision represented as [1, 1, 1, 1, 1].

[2]In the descriptions that follow, alternatives are referred to as p and q, attributes as A_i and A_j, and values of attributes for specific alternatives as $A_i[p]$. The symbols $\tilde{>}$ and $\tilde{<}$ indicates preference between two values, and $>$ and $<$ have their normal meanings.

3.1.2 Actions

The purpose of characterizing a decision on these dimensions is to reason about how to develop the decision dynamically. Decision making is viewed as a constructive process. Five general actions apply in the multiattribute, two-alternative model to help construct a decision:

- decision
- transformation by attribute
- transformation by importance
- substitution
- combination

Decision means selecting an alternative based on available evidence. For example, if importance distinguishes two attributes ($I[A_i, A_j] = 1$), then the alternative favored by the more important attribute is the decision. A decision can always be made, but with varying confidence. If you wish to increase confidence, another action should be performed instead.

The action **transformation by attribute** (abbreviated *Ta*) seeks to transform the current decision state by gathering information about one of the attributes. If the current information about an attribute is uncertain or unknown, this action attempts to resolve that uncertainty. The intent of a transformation is to change one decision state into another, hopefully more facilitative state. However, the desired transformation to a better state may not be possible; the actual transformation depends on the evidence obtained. For example, we may gather evidence about A_j with the hope of getting additional support for the currently favored alternative (supported by information we already have about A_i), but if the evidence, when obtained, indicates that A_i and A_j actually support different alternatives, then we end up in a state with a conflict.

The action **transformation by importance** (*Ti*) is like transformation by attribute but involves obtaining importance information.

We may wish to include other attributes. Two actions, substitution and combination, add new attributes. When one of the two existing attributes doesn't provide a significant difference, a new attribute may be **substituted** (*Su*) for it by discarding the existing insignificant attribute and replacing it with a new one.

Alternatively, we could **combine** (*Co*) the new attribute with the existing ones by taking advantage of the fact that there are only two alternatives. Since an attribute may only support one or the other of the alternatives, the attributes may be *clustered* together according to which alternative they support. Clustering is the key to extending a basic two-alternative, two-attribute representation to two-alternative, N-attribute representation, because it permits complex decision situations to be constructed iteratively within the framework of the decision typology.

These actions describe the state transitions that gather information and judgments about a decision and structure them such that the selection of an alternative emerges.

3.1.3 The Typology

Considering all the possible combinations of the values of the five dimensions, and pruning out isomorphic states (isomorphic with respect to the actions that may be taken) produces 24 basic states. Basic states have no clustered attributes.

The 24 states can be arranged in a table (Figure 1). The apples and oranges problem discussed earlier is state 23 in this table. Imagine that apples are preferred to oranges on flavor but oranges are preferred to apples on quality, and that flavor is the most important attribute. State 23 represents this situation as follows:

> There is a significant difference between *quality*[apples] and *quality*[oranges], so Sd[*quality*] = 1. Similarly, there is a significant difference between *flavor*[apples] and *flavor*[oranges], so Sd[*flavor*] = 1. There is a conflict (i.e., C[*quality*, *flavor*] = 1), implying that apples have better flavor but oranges have better quality, or vice versa. The attributes are not equally important (I[*flavor*, *quality*] = 1), in fact, flavor is preferred ($\tilde{>}$[*flavor*, *quality*] = 1).

Isomorphic states have been pruned out of the table. A full table would include 40 states. From the perspective of how a decision-maker acts, the 40 decision states contain some redundancies. Consider these states:

State 18a: $\mathrm{Sd}[A_i] = 1$, $\mathrm{Sd}[A_j] = 0$, $\mathrm{C}[A_i, A_j] = 1$, $A_i \tilde{>} A_j$
State 18: $\mathrm{Sd}[A_i] = 0$, $\mathrm{Sd}[A_j] = 1$, $\mathrm{C}[A_i, A_j] = 1$, $A_j \tilde{>} A_i$

In English, this state means

> "The dimension for which your evidence supports a decision is the most important dimension."

The states are identical in the sense that a decision-maker would not act differently in response to them. In both situations, the decision maker could try Ta, Su, or Co as reasonable next actions. Consequently, the two states are represented only by state 18 in the table.

Once the state of a decision has been identified, the table can be used to look up the set of possible actions. Figure 1 includes the appropriate actions for each state. The actions are divided into two rows. The first shows the actions for states with complete evidence. The second describes actions to be performed when some of the state information is missing.

Each of the actions has well-defined state transitions determined by the new information that they gather. Transformation is an appropriate action for any decision state with 0 in either of its first two rows or ? in its fourth. Substitution is appropriate when two alternatives are not differentiated on an attribute ($Sd[A_i] = 0$); since the attribute does not distinguish the alternatives, it should be replaced with one that does. Combination is appropriate anytime the decision is uncertain and more evidence should be gathered. This is typified best by states in which attributes support different alternatives (there is conflict) but have equal importance. The most appropriate actions (not all the possible actions) for a given state are listed in the table with numbers that indicate the set of possible destinations if the actions is performed. Note, as mentioned earlier, it is not possible to say exactly which of these states will arise until after the action is performed.

3.1.4 Effect of Actions on Decision State

Adding a new attribute via substitution and combination potentially affects every cell in a decision state, that is, each value $\mathrm{Sd}[A_i]$, $\mathrm{Sd}[A_j]$, $\mathrm{C}[A_i, A_j]$, $\mathrm{I}[A_i, A_j]$, and $\tilde{>}[A_i, A_j]$. In

State #		0	1	2	3	4	5	6	7
$Sd[A_i]$		0	1	1	0	1	1	0	0
$Sd[A_j]$		0	0	1	0	0	1	0	1
$C[A_i, A_j]$		0	0	0	1	1	1	0	0
$I[A_i, A_j]$		?	?	?	?	?	?	0	0
$\tilde{>}[A_i, A_j]$		*	*	*	*	*	*	*	*
Actions	All Info	Co D	Su D	D		Su,Co D	Co	Co D	Su D
	Part Info	Ta 0,1,4 Ti 6, 12,20	Ta 1,5,8 Ti 7, 13,14	Ti 5,8 21	Ta 3,4,5 Ti 9, 16,22	Ta 2,4 Ti 10, 17,18	Ti 11, 19,23	Ta 6, 7,10	Ta 7, 8,11

State #		8	9	10	11	12	13	14	15
$Sd[A_i]$		1	0	1	1	0	1	0	1
$Sd[A_j]$		1	0	0	1	0	0	1	1
$C[A_i, A_j]$		0	1	1	1	0	0	0	0
$I[A_i, A_j]$		0	0	0	0	1	1	1	1
$\tilde{>}[A_i, A_j]$		*	*	*	*	0	0	0	0
Actions	All Info	Co D	Su Co	Su Co	Co	Co Su	Su D	Su Co	Co D
	Part Info		Ta 9,10,7	Ta 10,11,8		Ta 12,13, 14,17,18	Ta 13,15, 17,19	Ta 14,15, 18,19	

State #		16	17	18	19	20	21	22	23
$Sd[A_i]$		0	1	0	1	0	1	0	1
$Sd[A_j]$		0	0	1	1	0	1	0	1
$C[A_i, A_j]$		1	1	1	1	0	0	1	1
$I[A_i, A_j]$		1	1	1	1	1	1	1	1
$\tilde{>}[A_i, A_j]$		0	0	0	0	1	1	1	1
Actions	All Info	Co Su	Su Co	Su Co	Co	Co Su	Co D	Co Su	Co
	Part Info	Ta 16,17, 13,14,18	Ta 17,19, 15	Ta 18,19, 15		Ta 20,13, 14,17,18		Ta 22,13, 14,17,18	

Figure 1: Basic Decision Typology with states and actions

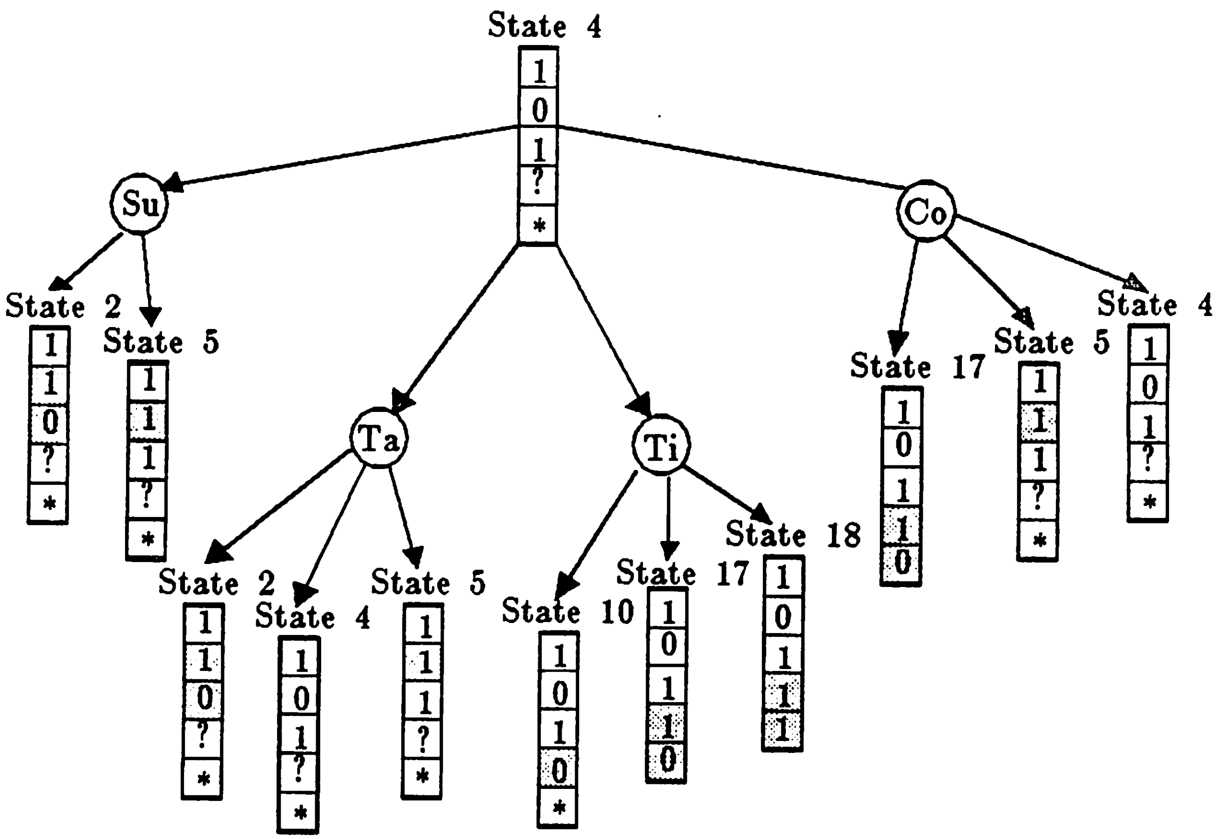


Figure 2: Single Transition with Multiple Attributes

combination with a new attribute, a previously insignificant single attribute may form part of a significant cluster (e.g., $Sd[A_i] = 0$ but $Sd[A_i \& A_{new}] = 1$). $\mathrm{C}[A_i,\ A_j]$ may change if the new attribute produces a conflict, and $\mathrm{I}[A_i,\ A_j]$ and $\tilde{>}[A_i, A_j]$ change simply by clustering attributes. Within the framework of the typology, the effects of adding a new attribute are:

1. to introduce a conflict where there was none
2. to take a side in a conflict
3. to join the consensus ($\mathrm{C}[A_i \& A_j, A_{new}] = 0$) but lend it legitimacy since $Sd[A_{new}] = 1$
4. to introduce an ordering where there was none (e.g. $\mathrm{I}[A_i, A_j] = 0$ but $\mathrm{I}[A_i, (A_j \& A_{new})] = 1$)
5. to change an ordering (e.g., $\tilde{>}[A_i, A_j] = 1$ but $\tilde{>}[A_i, (A_j \& A_{new})] = 0$

Figure 2 shows all the possible actions and their effects for a single state in the typology, state 4. In this example, there is enough of a difference to support a decision on A_i, but not A_j, and the evidence of the two attributes is contradictory. Four actions are appropriate: transformation by attribute (the 0 value for $\mathrm{Sd}[A_j]$ may indicate insufficient evidence), transformation by importance, substitution (for A_j), and combination. Note that it is possible to return to the same state, state 4, but by different paths. Substituting A_j or combining attributes transforms state 4 to state 5. But note that when state 5 was reached by combining attributes, one of them, A_i or A_j, actually represents the evidence of two attributes and so supports a decision more strongly.

Expanding the Definition of Attribute As presented in the typology, attributes refer to features of alternatives that are salient to the task of selecting the best alternative. This definition accommodates *outcomes*, *goals* or *characteristics*. Because each affects the decision differently and because we would like to be able to reason about the interaction of goals, the model has been extended to account for these separate types of attributes and how they interact.

Goal is the desired result (outcome) of a decision along a particular attribute.

Attribute is the actual result or repercussion of a decision, e.g., when you purchase a car, one result may be gas mileage, another may be reliability.

Expected outcomes are the expected results, with respect to a goal, of making a particular decision.

Characteristics are the actual attributes of the decision alternatives that contribute to the performance of those alternatives on each of the goals.

As an example, imagine the decision being between two cars, a Porsche 928 and a Nissan Maxima. Two goals in selecting a car may be fast acceleration and good gas mileage. The actual attributes of the alternatives might be fast acceleration for the Porsche and reasonable gas mileage for the Nissan. The expected outcomes are that the Porsche will be better on acceleration and the Nissan will be better on gas mileage. A characteristic of the car that is related to both attributes is engine size. So, the smaller engine size of the Nissan gives it better gas mileage, but slower acceleration.

To summarize, these four are related in the following way:

- A decision has many *attributes*.
- These *attributes* provide evidential support (by clustering as described earlier) to one of the two alternatives.
- A *goal* is a desired outcome along a particular *attribute*.
- Desired outcomes may be compared to *expected outcomes* on each attribute.
- An *expected outcome* for each alternative is a function of some subset of the set of *characteristics*. So expected outcomes form clusters of characteristics that lend support to the claim that a particular alternative performs better on a particular goal.

As can be inferred from their relationships described above, the four terms form a hierarchy (see Fig. 3). Refining the definition of attribute, in this way, indicates to the user more precisely the types of information being requested and expedites reasoning about interactions. In the description of the car selection, knowing that a characteristic, engine size, supports one goal, acceleration, and detracts from another, gas mileage, allows us to recognize a trade-off and attempt to seek other evidence that will distill its effect.

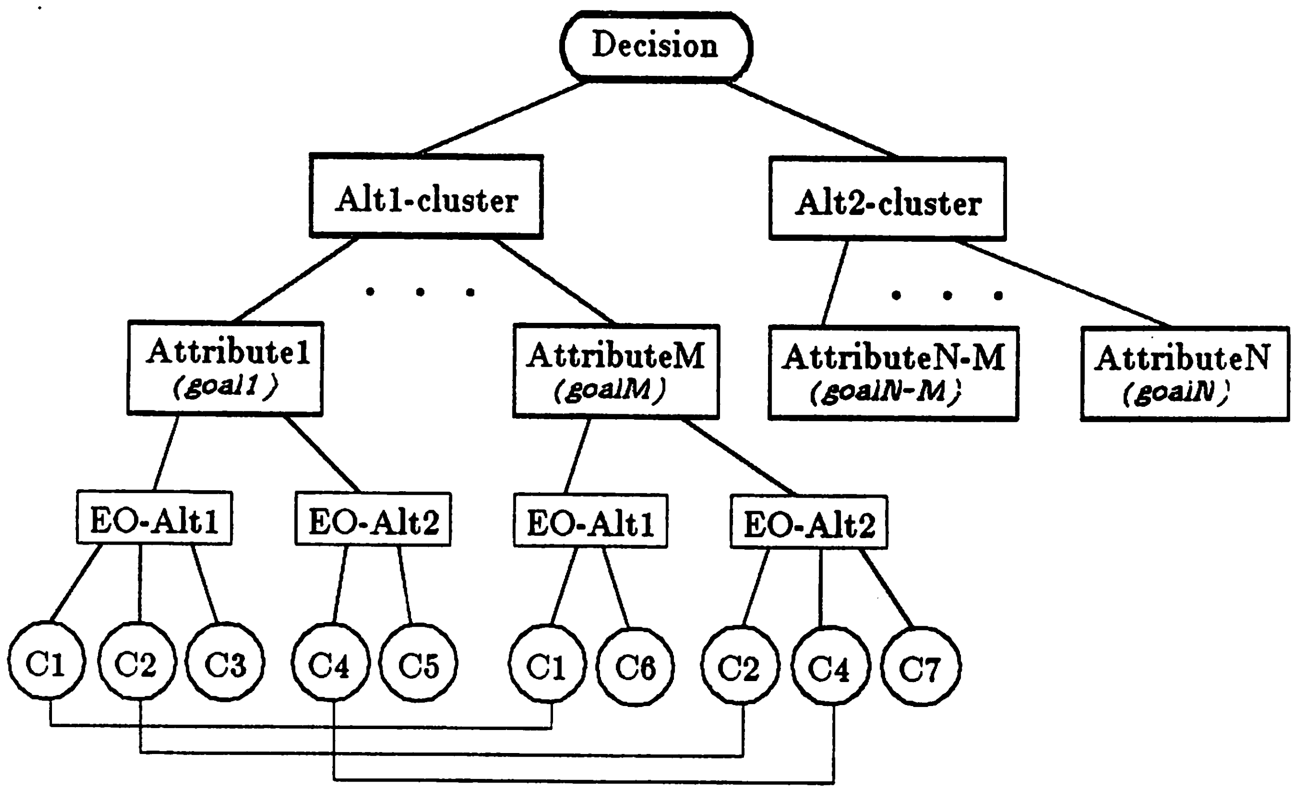


Figure 3: Extended Decision Hierarchy

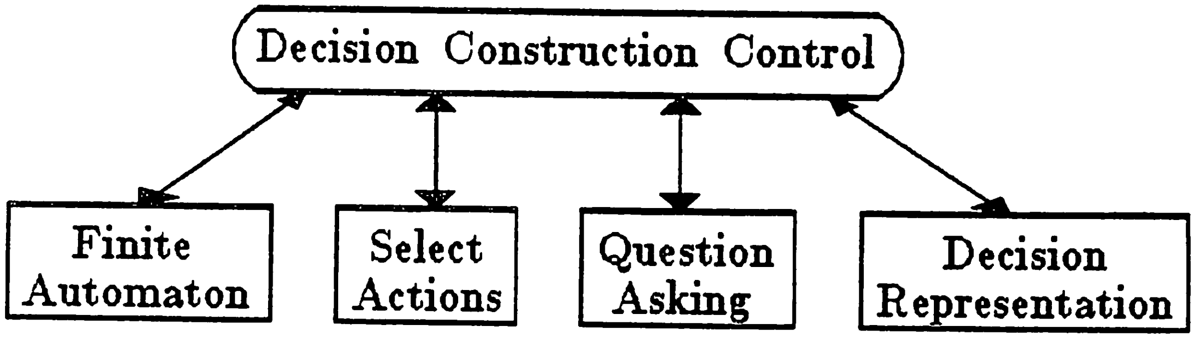


Figure 4: Program Modules

4 The CDM System: Program implementation

We have built a decision support system, called CDM, based on the decision typology[10]. CDM asks the user general questions about his or her decision, constructs a representation (as described in the previous section), selects actions to obtain additional information about the decision, and ultimately suggests an alternative.

CDM has five parts: a finite automaton to control state transitions, rules for selecting actions at each decision state, a question-asking interface, an internal decision representation, and a controlling program (Fig. 4).

The question-asking module and decision representation are fairly simple. The primary role of the question-asking module is to translate the actions to a form that the user can understand — direct questions that request the necessary information. This is implemented simply by mapping the actions to pre-determined questions. The internal decision representation keeps track of the state of the decision and provides routines to print and explain the decision state. The other modules implement the decision typology and the control of decision-making, and are described below.

4.1 Finite Automaton

The decision typology is represented as a finite automaton in which the states are the decision states described earlier and the arcs are labeled with the decision actions. The finite automaton module translates the dimensions of the typology into a state representation, determines the applicable actions and transitions for the state, and creates new states, as necessary.

Performing actions causes state transitions. The program determines which actions are applicable and predicts their possible results. For example, if we start from state 4 in Figure 2, the applicable actions are Su, Co, Ta, and To. These actions may result in states 2, 4, 5, 10, 17, and 18. The set of applicable actions for a given state are determined by following all applicable rules presented in Figure 5.

Each action has a set of rules for determining the possible changes to the dimensions and so the possible destination states. For transformation by attribute (Ta), getting attribute information causes the dimension Sd and possibly the dimension C to change. In transformation by importance (Ti), new states include changes to Importance and $\tilde{>}$. Because most of the

1. If the information about an attribute is unknown or uncertain (e.g., Sd=0), then suggest Transformation by Attribute.
2. If the relative importance of the attributes is unknown (e.g., I=?), then suggest Transformation by Importance.
3. If an attribute doesn't provide adequate support (e.g., Sd=0), then suggest Substitution.
4. If both attributes provide significant, corroborating evidence (e.g., Sd(A_i)=Sd(A_j)=1 and C(A_i, A_j)=0), then suggest Decision.
5. If at least one attribute provides significant evidence which doesn't conflict with the other attribute and it is considered to be more important (e.g., Sd(A_i)=1, C(A_i, A_j)=0, and $\tilde{>}(A_i, A_j)$=1), then suggest Decision.
6. Suggest Combination anytime.

Figure 5: Rules for determining possible actions

dimensions are determined relative to the attributes, performing a substitution changes Sd of the attribute being substituted, C, I and $\tilde{>}$. Combination actions conceivably change every dimension (in particular ways) because the new attribute gets combined with the existing ones.

Decision selections are made when necessary by evaluating the accumulated evidence. This evidence is gathered from the following rules that propose a selection.

1. If A_i has Sd=1 and A_j has Sd=0, then one can decide based on A_i.
2. If A_j has Sd=1 and A_i has Sd=0, then one can decide based on A_j.
3. If there is no conflict, then one can decide based on A_i.
4. If A_i is more important than A_j, then one can decide based on A_i.
5. If A_j is more important than A_i, then one can decide based on A_j.

4.2 Action Selection

Once CDM knows what actions are possible for the current state, it must select one. This is the task of the Action Selection module. Ideally, selecting the action should rely on domain dependent information or strategies promoted by the user. For example, a possible action may be to transform by attribute. In most cases, this would be the preferred action because it provides the most evidence; however, if the evidence is expensive, then the user may prefer to do something else. The current implementation does not include domain-dependent or user-dependent strategies for selecting actions, but relies instead on a simple, heuristic ordering of actions, augmented by some rudimentary reasoning about the decision states that may result from an action: at least one must be a state in which a decision is supported, otherwise some other action will be performed. Eventually, we intend for CDM to have a rule base and interpreter for implementing richer action-selection strategies.

4.3 Construction Control

The construction control module governs the interactions of the others. It initializes the system and directs the flow of information between the modules. The user interacts with the program through this module. Consequently, the module also interprets the user's responses and requests and directs the result to the appropriate submodule for further processing.

4.4 Results

The first version of the CDM program has been tested by several people. The user is presented with a decision and some facts about it, and uses the system to make a decision between two alternatives. The following transcript is an annotated version of one such session. The user was asked to select between two houses described in a real estate advertisement.

The following is a portion of a transcript of a decision making session using the decision typology. The system's responses are in `typewriter font`, the user's responses are in roman font, and annotations are in *italics font* for ease of identification. Portions that have been omitted without an explanation are indicated by vertical ellipsis. After every system action, the user may view parts of the growing representation. This option is presented after every action in a menu. Often that section of the transcript has been omitted to save space.

`Lisp>` (construct-decision)

```
Welcome to Version 1.0 of the Constructive Decision Support System.

The purpose of this program is to help you make a decision by asking you
questions about the alternatives, keeping track of your answers, and
trying to focus on the salient aspects of it. The system builds a sort
of pro/con representation to help explain why one alternative seems best.

The current version has a few restrictions:
    - the decision involves exactly two alternatives
    - all goals require at least two characteristics
    - once the program has selected an alternative, you can't override it

The program will ask you to evaluate characteristics and goals of your
decision. In answering these questions, you should consider ONLY the
attribute that you're being asked about. Don't evaluate the attributes with the
effects of others in mind. Try to select the attributes and think of them
independently of one another.

Finally, the program will ask you questions that refer to alternatives,
goals, characteristics, values and importance.

Would you like an explanation of these terms?
```
y

At this point, the program gives definitions of the basic terms: alternatives, goals, characteristics, values, and importance as they are described in the paper.

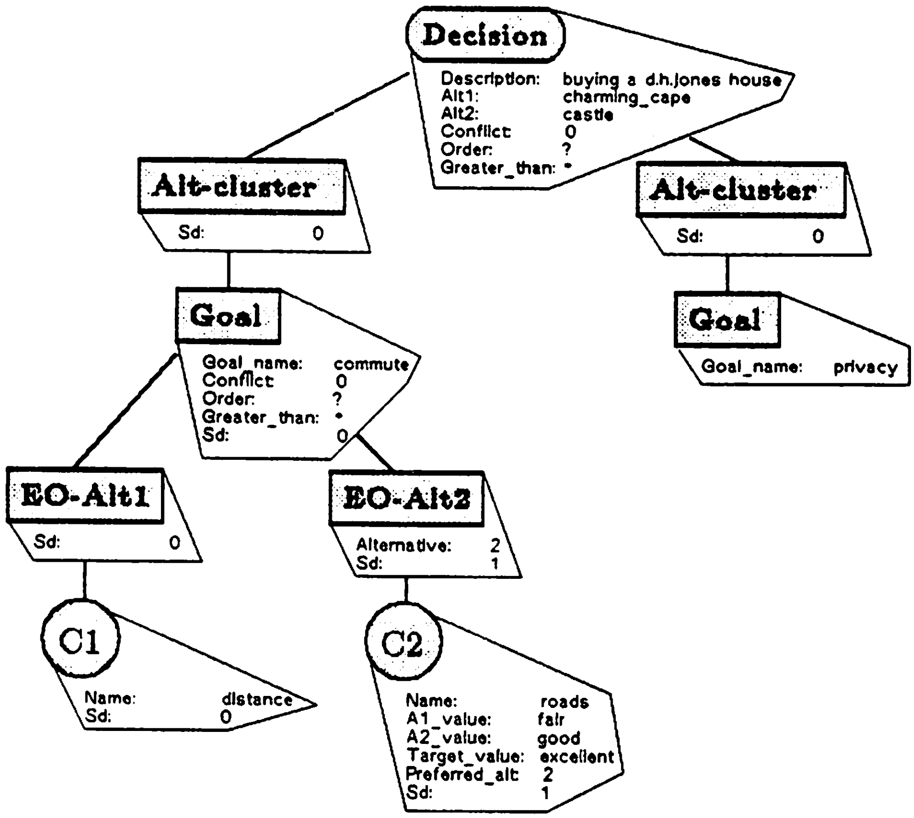
```
Please give a short text description of the decision problem:
 [empty line to end]
```
buying a d.h.jones house

```
What is the name of the first alternative?  [one word] charming_cape
What is the name of the second alternative?  [one word] castle
What is a goal of this decision?  [one word] commute
What is another goal of this decision?  [one word] privacy
What is a characteristic that affects the performance of the
    goal Commute?  [one word] distance
What is another characteristic that affects the performance
    of the goal Commute?  [one word] roads

Do you have information on the values of the alternatives on
    the attribute, Roads?  [y or n] y
What is the value of Charming_Cape for characteristic Roads? fair
What is the value of Castle for characteristic Roads? good
What is the best value that they can have? excellent
Is the difference between Charming_Cape and Castle
    significant on characteristic Roads? yes
Which alternative performs better on roads?
     [CHARMING_CAPE = 1 & CASTLE = 2, 0 = neither] 2

To examine all or part of the decision being constructed,
    select one of the following options:
      1  Print the decision tree
      2  Print a goal tree
      3  Explain the current state
      4  Break to lisp temporarily [type (continue) to return when finished]
      5  continue with the program

    which one? 1
```



The user has entered the basic information needed to start the system. The program used this information to build the tree displayed above by the user.

```
Do you have information on the values of the alternatives on
    the attribute, Distance?  [y or n] y
What is the value of Charming_Cape for characteristic Distance? half_hour
What is the value of Castle for characteristic Distance? half_hour
What is the best value that they can have? 10_mins
```

The system just performed a transformation by attribute. It had no information about the attribute, Distance, and so asked the user. In the section of the transcript that has been omitted here, the system performed a tranformation by importance to determine which attribute, Distance or Roads, is more important. Action selection is performed conservatively. Importance information is requested because it provides evidence used to distinguish the currently available characteristics should it happen that they are the only ones available. If other characteristics get included, the importance measure usually gets disregarded.

At this point, this system evaluates the available information (note: it now has 'complete' information about the characteristics it started with) with respect to making a decision and identifies a gap in the evidence: the DISTANCE characteristic doesn't really contribute any evidence to support either alternative.

```
One of the attributes doesn't contribute any evidence to the decision.
Is there another characteristic that is significant to the goal,
   COMMUTE? yes
What is its name? miles
Do you have information on the values of the alternatives on
   the attribute, Miles?  [y or n] y
What is the value of Charming_Cape for characteristic Miles? 10
What is the value of Castle for characteristic Miles? 20
What is the best value that they can have? 5
Is the difference between Charming_Cape and Castle
   significant on characteristic Miles? yes
Which alternative performs better on miles?
    [CHARMING_CAPE = 1 & CASTLE = 2, 0 = neither]  1
```

The program substituted the new characteristic, MILES, for the insignificant characteristic, DISTANCE. The new state that resulted from the substitution is in some ways less certain than the previous state. The old state permitted a choice because the only significant evidence supported alternative 2. Now however, one characteristic supports alternative 2 and the other supports alternative 1; so there is conflict that must be resolved.

```
Is one of the characteristics, (Miles) and (Roads),
more important than the other, with respect to COMMUTE? n
```

In an effort to resolve the conflict, the program requested importance information. In this case, there is no resolution based on importance.

```
Is there another characteristic that is significant to the goal,
   COMMUTE? y
What is its name? snow-route
```

In the omitted transcript, the program requested explicit information about the characteristic SNOW-ROUTE for the goal COMMUTE. Charming_Cape was on a snow route and the other wasn't. So Sd=1 and Charming_Cape is favored.

```
The characteristic, Snow-Route, is being combined with
   the other characteristics, Miles.
Do they together produce a significant difference?  [y, n or ?] y
:
Is one of the characteristics, (Miles Snow-Route) and (Roads),
   more important than the other, with respect to COMMUTE? y
Which is more important?  [0 = (ROADS), 1 = (MILES SNOW-ROUTE)] 1
```

Again following a conservative strategy, the program first requests importance information, as a fallback, and then checks to make sure that there isn't any other evidence.

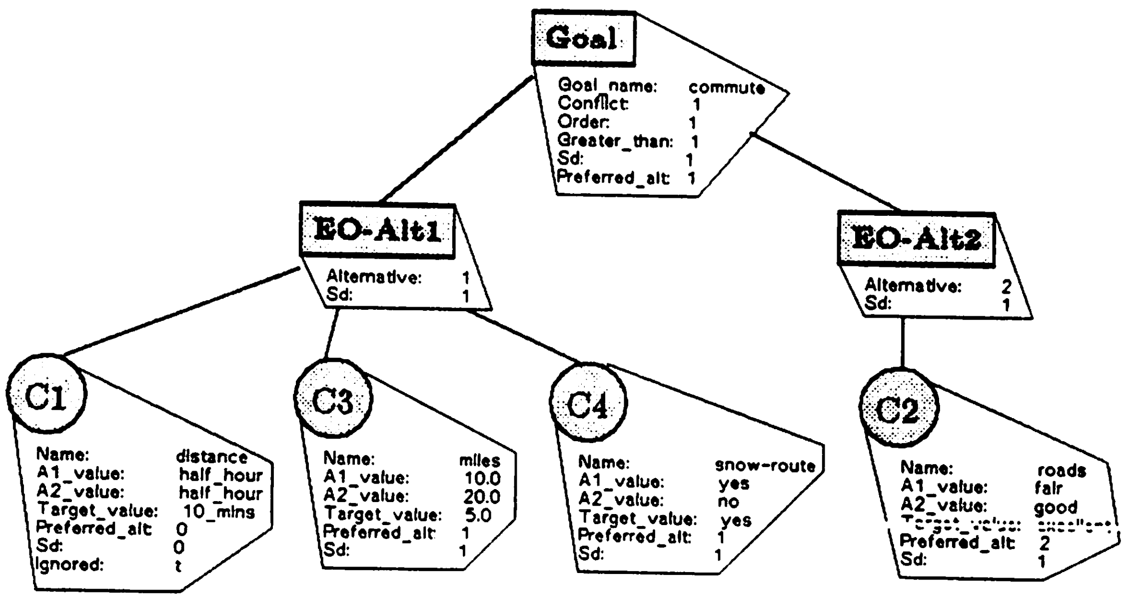
Since there is no more evidence, the goal, COMMUTE, is evaluated based on the available information and the program suggests a conclusion.

```
Is there another characteristic that is significant to the goal,
    COMMUTE? n
Given the set of characteristics, alternative Charming_Cape performs better
    on goal COMMUTE than does CASTLE.
Is that an acceptable conclusion?  [y, n, ?] y
Is the difference between the two alternatives significant on this goal? y

To examine all or part of the decision being constructed,
    select one of the following options:
      1  Print the decision tree
      2  Print a goal tree
      3  Explain the current state
      4  Break to lisp temporarily [type (continue) to return when finished]
      5  continue with the program

    which one? 2

Which goal would you like to see?
    1 COMMUTE
  goal: 1
```




```
To examine something else, select one of the following options:
      1  Print the decision tree
      2  Print a goal tree
      3  Explain the current state
      4  Break to lisp temporarily [type (continue) to return when finished]
      5  continue with the program
```

```
    which one? 3

The current configuration for the decision is [1, 0, 0, ?, *].

The decision can be described as buying a d. h. jones house.
Alternative 1 is called Charming_Cape. Alternative 2 is called Castle.
All evidence supports the same conclusion. The importance ordering for
the two clusters is unknown currently.
```

With the goal COMMUTE expanded, the other goal, PRIVACY, needs to be worked on. The program expands it in much the same way as COMMUTE. The user provides characteristics: proximity-to-highway and lot-size. The difference between the alternatives on LOT-SIZE is not significant.

```
One of the attributes doesn't contribute any evidence to the decision.
Is there another characteristic that is significant to the goal,
    PRIVACY? n
Given the set of characteristics, alternative Charming_Cape performs better
    on goal PRIVACY than does CASTLE.
Is that an acceptable conclusion?  [y, n, ?] y

Is the difference between the two alternatives significant on this goal? n
```

Now two goals have been expanded. Both COMMUTE and PRIVACY support alternative 1. COMMUTE is significant, but PRIVACY isn't.

```
One of the goals doesn't contribute any evidence to the decision.
Is there another goal that is significant to the decision? y
What is its name? pretty_house
```

The user is asked questions about the goal PRETTY_HOUSE. Its characteristics, EXTERIOR and HARDWOOD_FLOORS, support alternative 1. Its characteristic, FIREPLACE, supports alternative 2. The pair of characteristics are more important than the single one.

```
Is there another characteristic that is significant to the goal,
    PRETTY_HOUSE? n
Given the set of characteristics, alternative Charming Cape performs better
    on goal PRETTY_HOUSE than does CASTLE.
Is that an acceptable conclusion?  [y, n, ?] ?

The best decision seems to be Charming_Cape, alternative 1.
The attributes which support alternative 1 are preferred by importance ordering.
The current configuration for the decision is [1, 0, 0, ?, *].
The current configuration for the goal PRETTY_HOUSE is [1, 1, 1, 1, 1].

The goal PRETTY_HOUSE has the following values. There doesn't seem to
```

```
be a significant difference between the two alternatives on
PRETTY_HOUSE. The currently available evidence is conflicting - there
is some support for each alternative. The evidence in cluster 1 is more
important than that in cluster 2.
The following characteristics support alternative 1:
  exterior (significant), hardwood_floors (significant).

The following characteristics support alternative 2:
  fireplace (significant).

Is Charming_Cape an acceptable conclusion? y
Is the difference between the two alternatives significant on this goal? y
```

PRETTY_HOUSE gets substituted for PRIVACY. The program requests importance information and is told that PRETTY_HOUSE is more important than COMMUTE.

Another goal, SIZE, is added to the set. It has characteristics, BEDROOMS, BATHS, and GARAGE, supports alternative 2, and is significant. COMMUTE and PRETTY_HOUSE form a more important cluster.

```
Is there another goal that is significant to the decision? n


You can make a decision given the evidence that has been accumulated.
The favored alternative is CHARMING_CAPE.
Is that an acceptable selection? [y, n, ?] ?

The best decision seems to be CHARMING_CAPE, alternative 1.
The attributes which support alternative 1 are preferred by importance ordering.
The current configuration for the decision is [1, 1, 1, 1, 0].

The decision can be described as buying a d. h. jones house.
Alternative 1 is called Charming_Cape. Alternative 2 is called Castle.
The currently available evidence is conflicting - there is some support
for each alternative. The evidence in cluster 1 is more important than
that in cluster 2.
Alternative 1 is supported by goals: commute (significant),
  privacy (insignificant), pretty_house (significant).
Alternative 2 is supported by goals:
  size (significant).

Is CHARMING_CAPE an acceptable selection? y
```

"Decision construction completed"

Since there are no more goals, the program recommends an alternative as the decision and gives the user an opportunity to review the evidence supporting it. If *it is acceptable, the process ends. Otherwise, the program tries with the user's help to revise the decision.*

The transcript illustrates the flow of control as guided by the topology. For example, on page 19, the system recognizes that one of the attributes, previously thought to be significant, DISTANCE, was not and so should be replaced by something else. This action takes it from state 7, configuration [0, 1, 0, 0, *], to state 5, configuration [1, 1, 1, ?, *] in which there is more evidence, but as it happened, less information to distinguish the alternatives. After three attributes have been gathered to support the goal COMMUTE, the program suggests that the alternative, Charming_Cape, performs better.

Because the emphasis was placed on the style of reasoning, rather than the user interface, the interaction is a bit rough. Future versions of the program will include an improved interface with better explanations and some form of sensitivity analysis for allowing the user to consider the repercussions of uncertain judgments. Additionally, as the mechanism is enhanced to include multiple alternatives, the conflict resolution between actions will become correspondingly more sophisticated.

5 Conclusion

In the CDM approach, the evolution of a decision is viewed as a sequence of transitions through a state space. The 24 basic states represent all comparisons between two alternatives on two attributes, and also represent comparisons based on clusters of attributes. A decision can be made in any state; however, the user can have more confidence in decisions made from some states than others. For example, state 5 in Figure 1 represents a situation in which one attribute supports the first alternative and another supports the second alternative. A decision *can* be made in this situation, but it would be arbitrary because it lacks information about the relative importance of the attributes. In most cases, the decision-maker will want to transform state 5 with the Ti operator into a state in which the relative importance of the attributes is known. Three possible states may result from asking for this information: states 19 and 23 represent the preference of one attribute or the other, respectively, and state 11 represents the case in which the user says he or she has no preference between the attributes. Because search operators such as Ti gather information from the user, one cannot predict in advance which state will result from their application: one applies Ti hoping to transform state 5 into state 19 or 23, but one may end up in state 11.

The remainder of this section offers a comparison between constructive decision making and decision analysis, then discusses extensions to the approach.

5.1 Constructive decision making and decision analysis

Constructive decision making contrasts with decision-theoretic approaches in several ways. A summary is shown in Figure 6[3]. Decision analysis guarantees optimal decisions with respect to

[3]This table was produced with help from Tammy Tengs, a member of the department of Operations Research at UMass.

	Decision Theory	Decision Typology
Goal	optimizing	satisficing
Algorithm	combines evidence	gathers evidence
Evaluation	static	dynamic
# of alternatives	many	2
Comparison scales	single	multiple
Informational burden on user	reductionistic	holistic
Ignorance of attribute values	assume some distribution	disregarded or deferred
Ignorance of attribute importance	assume equality	disregarded or deferred
utility theory	important & explicit	not explicit
numeric/symbolic	probabilities	reasons

Figure 6: Comparison of Decision Theory and Decision Typology

a model by dividing decision-making into two phases, one in which the model (e.g., a decision tree) is constructed, and another in which the best alternative is determined. Decision analysis gives a formal accounting of how evidence is combined and how the subjective expected utility of outcomes is calculated, but it does not explain how the components of a model, including the relevant evidence and salient outcomes, are identified. In contrast, constructive decision making is a satisficing approach because at any point in the process the user can request the decision that is currently indicated by the evidence. If the process continues, the decision might change. Constructive decision making gives a formal accounting of how evidence is gathered; its methods for combining the evidence are relatively weak and are not the main emphasis of the approach.

Decisions offered by constructive decision making may be wrong according to the criteria of decision analysis, but these criteria are not appropriate for constructive decision making. Viewing both approaches in terms of search clarifies this point: A decision tree is a static search space, typically small, that can be searched exhaustively (e.g., by the "averaging out and folding back" method[18]) for the best alternative. The search space for constructive decision making is infinite because one or more actions is possible in every state of the decision typology. One can never find the best decision, only the one that is currently favored by the evidence. Since there is no guarantee that more evidence will not change the decision, constructive decision making systems must be evaluated on how the process of constructing decisions is controlled. Decision analysis avoids this issue by assigning the analyst the responsibility of limiting the evidence and outcomes that will be considered, then searching for the best alternative in this limited space.

Both approaches make assumptions about how decisions will be represented. Several limitations of the representations for constructive decision making are discussed below. The most serious may be that it is limited to two-alternative situations, whereas decision analysis can handle more alternatives. An assumption of decision analysis is that outcomes can be compared on a single utility scale (although there are multiattribute decision theory methods). Constructive decision making was developed partly in response to the limitations imposed by this assumption. In decision analysis, the user assesses absolute worth of individual attributes on a single scale. Thus, the theory performs the combination. In CDM, the user is not required to map worth to a single scale, but is required to perform the combination when the contributions of the attributes pro and con the alternatives are combined to produce a decision. So, the difficulty of mapping to utility is traded against making comparisons of clusters of attributes. Additionally, the final result of the combination for decision theory is numeric, combinations of probabilities and utilities; whereas, for construction decision making, the result is symbolic, reasons for support.

5.2 Further Research

We are considering three kinds of extensions to the constructive decision making approach. First, the typology can be modified to differentiate decision situations that are currently undifferentiated. Second, the actions associated with the states in the typology can be augmented. Third, control of these actions should be more sophisticated.

Augmenting the typology to better differentiate decision situations Each state in the current decision typology represents four or five qualitative judgments: Are the alternatives significantly different on attribute-1? on attribute-2? do these attributes favor different alternatives? is one attribute preferred? if so, which one[4]? The typology does not represent the extent of the difference between alternatives on an attribute, the degree of conflict between the alternatives, or the weights of the attributes. Thus, it cannot differentiate decision situations that depend on these quantitative judgments, situations that intuitively seem different. For example, in some cases the typology recommends an alternative that quantitative judgments suggest is inferior.

This situation can arise if we force a decision in state 23 of the typology, in which the alternatives conflict on features i and j, but feature i is preferred to feature j. The action associated with state 23 is *Co*, suggesting that other attributes should be considered. But decisions are possible in any state of the typology, albeit with varying confidence. In state 23, the best decision is the alternative that wins on attribute i, since i is the preferred attribute. Now imagine that feature i is k times better than feature j. One interpretation of this statement is that we would trade k units on j for one unit on i. For example, we might be willing to add 5 miles to a commute for each increment in privacy of our house — in which case distance

[4]Throughout this section we use the term attribute to refer to single attributes and to attribute clusters.

has 1/5th the weight of privacy, and privacy is in some sense 5 times better than distance[5] However, if the difference between the alternatives on the less-preferred attribute is more than k times greater than the difference between the alternatives on the preferred attribute, then we would probably select the alternative favored by the less-preferred attribute. Concretely, if $house_1$ involves a five-mile commute and has a privacy score of 2, and $house_2$ involves a 25-mile commute and has a privacy score of 4, then we might prefer $house_1$, because $house_1$ is equivalent to a hypothetical $house_3$ that involves a 15-mile commute and has a privacy score of 4, and $house_2$ is clearly inferior to $house_3$. Contrary to this argument, the typology recommends $house_2$ because it dominates $house_1$ on privacy, the preferred attribute[6].

A second extension to the typology is needed to represent probabilistic statements about the outcomes of choices. Currently, we can say one alternative is preferred (on an attribute) but not that one alternative is likely to be preferred. We have no way to represent a probability distribution of, say, privacy scores given each alternative. This is an easy problem to fix: we could in fact acquire such probability distributions for each alternative and interpret the Sd dimension of the typology in terms of overlap between the distributions[24].

The most severe limitation of the current typology is that it only represents two-alternative decisions. If we wanted to keep the current typology but select among three houses, we would have to do it by pairwise comparisons. Whether we should do this or redesign the typology for multiple alternatives is a question we are currently exploring. The question highlights a tension between representation and control that we see in CDM and most other AI systems. Rephrased, the question is whether one's representations support a desired range of problem-solving strategies. The current representation of decision states — the typology — may be sufficient to support the strategy of making multi-alternative decisions by pairwise comparisons, but it is not sufficient to, say, represent several clusters of several alternatives on a single dimension. The decision to redesign or augment a representation depends on whether the strategies supported by that representation are sufficient to solve a problem efficiently. At this point, we do not know whether multi-alternative decision making warrants redesigning the two-alternative typology.

Extending the typology by adding actions Each state in the current typology has one or more actions associated with it. There are five actions in all. Two gather information about attributes and importance (*Ta*, *Ti*), respectively; two add new attributes (*Su*, *Co*); and one, ***decision***, involves selecting an alternative. If the typology is extended as discussed above, then additional actions may be required. For example, a multi-alternative constructive decision maker will probably require actions to cluster alternatives, much as attributes are clustered by the *Co* action.

Perhaps the best way to integrate constructive decision making with decision analysis is by adding evidence-combining actions to the typology. Evidence combination is currently very simplistic: attributes are clustered according to the alternatives they support and the preferred

[5] Tradeoffs are not the only interpretation of feature preference. Moreover, they are generally nonlinear; for example, the marginal gain in privacy from "very private" to "extremely private" may not be worth five additional miles driving. Nonetheless, this interpretation of feature preference will serve to illustrate the limitations of purely qualitative judgments in the typology.

[6] We must stress that this example assumes a decision is forced in state 23, when in fact the typology suggests *Co* in that state.

cluster (I$[A_i,A_j]$) is identified. In some cases, we add up the number of attributes that support a cluster. The typology should be augmented with actions to combine evidence in more sophisticated ways.

Augmenting control strategies for constructive decision making The typology associates decision situations with actions but it does not select which of the applicable actions to take. The task of controlling actions in the typology has two aspects. First, the actions associated with states are those that seemed best to the designers of the typology, although most actions can be applied in most states. For example, the actions listed for state 5 are Co and Ti, but Su and Ta are also applicable, and *decision* is always applicable. Notably, Ti is applicable only in states 0 ... 6 and is redundant elsewhere. But with this exception, actions are more widely applicable than the typology suggests (e.g., Co is applicable in every state). When an action is not associated with a state (e.g., *decision* in state 5) its absence represents some implicit control knowledge. For example, the absence of *decision* in state 5 represents a judgment on the part of the designers of the typology that decisions are inappropriate when there is conflict but the relative importance of the attributes is unknown. This implicit knowledge could (and probably should) be made explicit in the action-selection component of CDM.

Whereas the actions associated with states in the typology provide implicit control, a second aspect of controlling actions is explicit, dynamic conflict resolution. Irrespective of whether the typology recommends all actions applicable in a state or some predetermined subset, it will still be necessary to choose among recommended actions. For example, all five actions are possible (and recommended) in state 4. Which should be taken? Many extraneous factors seem pertinent: if the cost of evidence is very high and the consequences of a bad decision are relatively minor, then *decide* might be the best action; in the opposite case, if Sd$[A_j]$ is unknown, then Ta is appropriate; alternatively, we might decide to seek corroboration for the alternative favored on attribute i, in which case Co is appropriate. Constructive decision making, and the CDM system in particular, requires knowledge to select actions. Stated positively, the control of constructive decision making is entirely up to the system builder. A few simple rules, such as those shown in Figure 5, will suffice to make CDM run, and there are hooks in CDM for more complex control strategies. If, for example, CDM is used to make decisions in real-time problem solving, then one strategy might be to obtain an initial decision as quickly as possible but defer it, acquiring more evidence (perhaps replacing old evidence with more recent, timely evidence using Su) if time permits.

Strategies for constructing decisions can be complex and domain-specific. The typology should not bias the behavior of a decision-making system such as CDM. For this reason, the typology should recommend all possible actions, and allow domain-specific strategies to select among them. Currently, the typology does not represent aspects of decision situations that are required to support complex strategies; for example, because the typology represents states in terms of qualitative judgments, its best strategy for making a decision in state 23 is to select the alternative that dominates on the preferred attribute. That this conflicts with recommendations based on quantitative judgments is neither surprising nor an indictment of the decision-making strategy for the qualitative typology. If the typology represented quantities, we would identify the best alternative on quantitative criteria; since it does not, we select the best alternative qualitatively. The important points are that this version of the typology — and all future versions — should maintain the distinction between the states of a decision and the potential actions in each state; and that each state should represent enough information about the decision situation for the decision-making system to select the best action that applies in a state.

Finally, CDM is intended to explore methods for structuring decision problems, comparing alternatives on their attributes, reasoning about the state of decisions, and automating dynamic decision-making. It is not designed to produce optimum solutions given complete information, but rather to help us understand how decisions are constructed.

References

[1] Avron Barr and Edward A. Feigenbaum, editors. *The Handbook of Artificial Intelligence*, chapter 5. Volume 1, William Kaufmann, Inc., 1981.

[2] Stanley F. Biggs and Theodore J. Mock. An investigation of auditor decision processes in the evaluation of internal controls and audit scope decisions. *Journal of Accounting Research*, Spring 1983.

[3] Paul R. Cohen, David Day, Jefferson Delisio, Michael Greenberg, Rick Kjeldsen, Daniel Suthers, and Paul Berman. Management of uncertainty in medicine. *International Journal of Automated Reasoning*, Spring 1987.

[4] Paul R. Cohen, Glenn Shafer, and Prakash P. Shenoy. Modifiable combining functions. *AI EDAM*, 1987.

[5] F.Nelson Ford. Decision support systems and expert systems: a comparison. *Journal of Information and Management*, 8:21–26, 1985.

[6] Barbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26, 1985.

[7] Max Henrion and Daniel R. Cooley. An experimental comparison of knowledge engineering for expert systems and for decision analysis. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, AAAI, 1987.

[8] R. A. Howard. Decision analysis: applied decision theory. In D.B.Hertz and J.Melese, editors, *Proceedings of the Fourth International Conference on Operational Research*, pages 55–71, Wiley, New York, 1966.

[9] R. A. Howard. The foundations of decision analysis. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4:211–219, Sept 1968.

[10] Adele Howe. *A Constructive Approach to Decision Making*. Master's thesis, University of Massachusetts, 1986.

[11] D. Kahneman, P. Slovic, and A. Tversky, editors. *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge University Press, Cambridge, 1982.

[12] Antonio Leal and Judea Pearl. An interactive program for conversational elicitation of decision structures. *IEEE Transactions on Systems, Man and Cybernetics*, 7(5), May 1977.

[13] R. Luce and H. Raiffa. *Games and Decisions*. Wiley and Sons, Inc., New York, 1957.

[14] D. W. North. A tutorial introduction to decision theory. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4(4), Sept 1968.

[15] J. Payne, M. Braunstein, and J. Carroll. Exploring predecisional behavior: an alternative approach to decision research. *Organizational Behavior and Human Performance*, 22:17–44, 1978.

[16] John W. Payne. Contingent decision behavior. *Psychological Bulletin*, 92(2):382–402, 1982.

[17] J. Pearl, A. Leah, and J. Saleh. Goddess: a goal-directed decision structuring system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4:250–262, 1982.

[18] Howard Raiffa. *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*. Addison-Wesley, Reading, MA, 1970.

[19] A. Sage and C. White. Ariadne: a knowledge-based interactive system for planning and decision support. *IEEE Transactions on Systems, Man and Cybernetics*, 14:35–47, 1984.

[20] C.A.S. Stael von Holstein. A tutorial on decision analysis. In *Third Research Conference on Subjective Probability, Utility, and Decision Making*, London, Sept 1971.

[21] O. Svenson. Process descriptions of decision making. *Organizational Behavior and Human Performance*, 23:86–112, 1979.

[22] Detlov von Winterfeldt and Ward Edwards. *Decision Analysis and Behavioral Research*. Cambridge University Press, 1987.

[23] Michael Szu-Yuan Wang and James F. Courtney, Jr. A conceptual architecture for generalized decision support system software. *IEEE Transactions on Systems, Man and Cybernetics*, 14(5), September/October 1984.

[24] Robert L. Winkler. *Statistics...*