

Decision Classification Enhances Case-Based Reasoning

Stephen L. Woehrle, (E-mail: woehrs@mnsu.edu), Minnesota State University of Mankato

Abstract

In order to arrive at a meaningful and useful classification of decisions, a sufficiently descriptive framework for classification should be created. The use of the traditional programmed-nonprogrammed dichotomy is acceptable, but decisions themselves cannot always be treated as a single either/or entity. Decision-making is a process involving distinct and separable steps. A programmed-nonprogrammed classification scheme will be applied to a variety of different types of decisions. Using Case-based Reasoning (CBR) various decision scenarios with their corresponding classifications can be stored. The resulting database is a growing and evolving body of experience and knowledge to be used for future decision-making.

Introduction

Since their christening by Herbert Simon, the terms “programmed” and “nonprogrammed” have been widely used for classifying decisions. Programmed decisions are highly repetitive or routine decisions for which definite procedures have been developed. Decisions are nonprogrammed to the extent they are novel with no specific method for dealing with them. This classification involves a continuum with only the extremes representing the two distinct polar types.

With ever-increasing amounts of information and with a myriad of decision techniques available, an accurate accounting of the decision-maker’s function becomes more important. For frequently occurring decisions, top management should design definite policies or procedures, which permit delegation of these matters to subordinates or computers. Delegation requires that managers know which problems frequently occur and what techniques are applicable for routine solution. A concern for decision classification can also be justified on a cost basis. The cost of general-purpose techniques for decision-making is usually high and should be reserved for nonprogrammed decisions for which no routine applications are available. For often-encountered problems, special-purpose methods can be designed to give less costly and perhaps better solutions.

In order to arrive at a meaningful and useful classification of decisions, a sufficiently descriptive framework for classification should be created. The use of the traditional programmed-nonprogrammed dichotomy is acceptable, but decisions themselves cannot always be treated as a single either/or entity. Decision-making is a process involving distinct and separable steps.

The purpose of this paper is twofold. The first purpose is to develop a 4-stage programmed-nonprogrammed decision-making classification scheme. The second purpose is to apply Case-based Reasoning (CBR) to the classification scheme in order to promote a higher degree of programmed decision-making.

A programmed-nonprogrammed classification scheme will be applied to a variety of different types of decisions. Using Case-based Reasoning (CBR) various decision scenarios with their corresponding classifications can be stored. The resulting database is a growing and evolving body of experience or knowledge to be used for future decision-making.

Readers with comments or questions should contact the author via email.

Decision-making Process

There exist many definitions of the decision-making process. Perhaps the most favorable ones adhere to the scientific method, which requires an initial definition of the problem and the subsequent selection of a best alternative following careful evaluation of various possibilities. The decision making process may be analyzed in four separate steps. These are:

1. Problem identification
2. Search for alternatives
3. Evaluation of alternatives
4. Choose an alternative and initiate action

These stages are implicit in all levels of decision making, although they may receive different emphasis. Problem identification, the starting point for decision-making, determines which decisions are to be undertaken. The initial selection of areas of concern or attention to pertinent problems may be the essence of management. Problems come to the attention of the decision-maker in a variety of manners. Some problems, such as budgets, are encountered on a regular schedule; others arise automatically from detection by control devices; and still other problems precipitate as underlying causes following periods of investigation. Problems such as personnel vacancies can arise unannounced, while problems relating to objectives may be carefully planned and developed.

The development of alternative courses of action is the second phase of decision-making. While some problems are limited to a rather obvious "either-or" type solution, as in the case of a make or buy decision, the majority of problems can make use of creativity and imagination in developing a range of alternative choices. To arrive at realistic alternatives, at some point constraints must be recognized which are peculiar to the problem situation.

The third step in the decision-making process involves evaluation of alternatives. The collection of relevant data and the development of meaningful information become part of the analysis. Due to imperfect information and uncertainty, the probable outcomes of each solution must be estimated. The decision-maker establishes criteria to determine which alternative best optimize desired results in terms of objectives. To instill a degree of consistency and objectivity in evaluation, many criteria have been operationally defined and expressed in quantitative terms. Probability theory, to assess risk, and linear programming, for optimal resource allocation, are two such quantitative techniques in a large arsenal. It should be noted that inputs into the objective techniques and criteria, which do not lend well to quantification, are inevitably dependent on subjective judgment. Specific criteria sets and evaluation techniques can be designed for application to particular re-occurring problems and more general evaluation programs can be developed for given problem types.

The final stage in decision-making entails the actual selection of an alternative following the process of evaluation. The decision-maker should consider all the information generated from previous analyses and should also utilize his/her own insight. Once a course of action has been decided upon, it is of equal importance that an operative initiating structure exists. This may entail forward communication, a mechanism for enacting the solution, and a commitment to the decision.

Decision-classification Schemes

Once the decision making process has been defined in a series of steps, the programmed-nonprogrammed dichotomy can be employed for each stage to yield a more descriptive classification scheme.

Initial Classification

The decision-making process can initially be divided into two main functions; problem identification and subsequent solution. Identification, as previously discussed, relates to determining the problem (Step 1), while solution denotes the remainder of the decision-making process (Steps 2-5).

A two-dimensional taxonomy may be used to describe the degree to which decisions are programmed or nonprogrammed with respect to problem identification and solution. See Exhibit 1.

Exhibit 1
Problem Identification and Solution: Programmed or Nonprogrammed

Solution \ Identification	Programmed	Nonprogrammed
Programmed	I	II
Non-Programmed	III	IV

Programmed identification indicates that problems emerge in a routine or automatic manner, while nonprogrammed determination would relate more to the seeking out and defining of problem areas. It should be noted that once new problems have been identified they could be placed on a regular schedule or caused to occur automatically through the use of control devices.

Referring to Exhibit 1, Cell I problems may be both routinely identified and solved. This represents an almost entirely quantitative approach to decision-making. For example, inventory maintenance problems can involve the application of reorder points and standard lot size formulas. The problem of additional inventory required would automatically emerge under a periodic or perpetual inventory system, the periodic system being time dependent and the perpetual design being event triggered. In both situations, the alternative quantities could be routinely calculated by reorder formulas and requisitions could be made by peripheral equipment. Cell II problems display routine identification and novel solution. An example of this situation is the use of quality control charts for on-going processes. Such monitoring devices indicate when a process is out of control, but they do not, in general, define the cause or supply a remedy.

Cell I and II are most often related to the control function, and are indicative of operating level decisions. Special solutions to routine problems, if successful, are typically made routine in the form of policies or procedures.

Problems denoted by novel identification and routine solution pertain to non-recurring situations that are solved with systematic applications. Capital budgeting models making use of present value analysis to deal with major acquisitions, or the use of linear programming to determine optimum mix, would be representative of Cell III types of problems.

The setting of policies and objectives indicates a problem type with special identification and special solution, Cell IV. No standard approach to problem determination and solution is available. Cell III, and especially Cell IV, relate to planning and emphasize a more qualitative approach to decision making.

Secondary Classification

A more detailed classification can be accomplished by further segmenting the solution function into its three previously discussed steps: alternative search, evaluation, and choice. See Exhibit 2.

Exhibit 2
The Decision Making Process: Programmed or Nonprogrammed

Stage of Process	Programmed	Nonprogrammed
Problem Identification	A	E
Search for Alternatives	B	F
Evaluation of Alternatives	C	G
Choice of Alternative and Initiating Action	D	H

This refinement is necessary because few decisions have all stages of solution entirely programmed or non-programmed. A cybernetic system entailing automatic feedback and control would involve completely programmed decisions. This programmed homogeneity in all stages is represented by an ABCD classification in Exhibit 2. Both a programmed search for alternatives and their evaluation indicates that the plausible alternatives based on constraints and the criteria for evaluation have already been considered and represented in a routine procedure or technique.

An inventory reorder problem could also be ABCD if it utilizes reorder point, economic order quantity, and requisition models. The same inventory reorder problem could be EFGH if no previously developed models have been implemented.

An EFCH designation could describe a nonrecurring aeronautic design problem for which many alternative ideas are contributed through brainstorming sessions. Suggestions may be subjected to a standard set of criteria relating to aerodynamics and economics. The final choice, if not entirely determined by the variables evaluated, can be novel to the extent that it includes individual preference. This preference may evolve from interaction, insight, or even aesthetics. An EFGH designation might relate to strategic planning which involves novel expressions at each stage of the process.

The decision-maker should attempt to develop routine procedures for each stage of problem he encounters. Programmed applications for each stage of a particular problem can be sought from previous experience, company/industrial information and/or existing literature. The development and utilization of such a database can be accomplished by using case-based reasoning (CBR).

Case-based Reasoning

Expert systems primarily capture the knowledge of individual experts. But organizations also have collective knowledge and expertise that they have built up over time. This organizational knowledge can be captured and stored using case-based reasoning. Case-based reasoning solves new problems by using or adapting solutions that were used to solve old problems. The goal of case-based reasoning systems is to be reminded of similar cases when faced with a new problem and to retrieve those cases in order to help the user solve the new problem in a fashion similar to the way it was solved for a closely matching situation. In case-based reasoning, descriptions of past experiences of human specialists, represented as cases, are stored in a database for later retrieval when the user encounters a new case with similar parameters. The system searches for stored cases similar to the new one, finds the closest fit, and applies the solutions of the old case to the new case. Case-based reasoning retains the rich context of a

case and this has advantages over the distilled knowledge associated with rule-based systems.

Case-based reasoning is an attempt to relieve the knowledge acquisition bottleneck by replacing the knowledge of an expert with a database of historical *cases*. These cases are examined for their significant features, and, based on their resemblance to the present problem; their solutions are used to generate a solution to the present problem.

Case-based reasoning represents organizational knowledge as a database of cases that can be continually expanded and refined. When the user encounters a new case, the system searches for similar cases, finds the closest fit, and applies the solutions of the old case to the new case. The new case is stored together with successful solutions in the case database.

Human problem solving is based somewhat on the application of experiences to the current problem. Experts become experts by doing and experiencing. In heuristic systems an expert's knowledge is represented as compiled knowledge that is, rules of thumb, shortcuts, etc. It is unlikely that when asked, an expert will be able to recall the details of more than a few problem-solving cases. Thus the experience represented is compiled, rather than explicit. When a rule-based diagnostic system is developed, its rules represent a distillation of knowledge. The original reasons why a rule was created may become obscure with time. However, keeping the entire case allows one to follow procedures that had proved successful in a prior situation. Also, in a rule-based system one might have to prove that a rule is theoretically correct. This could be a very complex problem.

On the other hand, the concept of case-based reasoning is founded on the idea of using explicit, documented experiences to solve new problems. The entire case is available, not just distilled information. Expert systems work by applying a set of IF – THEN – ELSE rules against a knowledge base where both the rules and the knowledge base are extracted from human experts. In contrast, case-based reasoning represents knowledge as a series of cases and this knowledge base is continuously expanded and refined by users. Whereas expert systems have to be continually revised by incorporating new rules or modifying existing ones, case-based reasoning systems can be expanded by the addition of new cases.

A case-based reasoning system consists of three basic components:

1. A library of historical cases
2. A means of using key elements of the present problem to find and retrieve the most similar cases from the case library
3. A means for making modifications to the proposed solution when the case on which the solution is based is not identical with the current problem

The construction and organization of a library of cases is a critical task. Inefficient searches through the case library due to its poor organization can result in unacceptable performance. The case library may be organized as a flat database or hierarchically based on some organizational structure with inputs or goals. Furthermore, the cases in the library must be indexed to allow for efficient searches.

Building a case-based reasoning system involves the development of a fairly robust and well-indexed set of cases. Ideally, the case library would be composed of documented historical cases. Case-based reasoning is an excellent technique to use when many well-documented histories of past problems and their solutions exist. For example, the legal profession thrives on judicial precedent, where a decision about a current case can be dependent on a landmark decision on a similar case.

Some advantages of case-based reasoning over the more traditional deductive rule-based systems are:

1. Case-based reasoning is patterned after human reasoning. While rule-based systems depend on the prior experience of an expert, their knowledge is represented quite differently. A case library is a more formal

- and explicit representation of this knowledge, since it details the case history of the domain directly rather than through the interpretation and recollection of an expert. The problems faced before will be compared to the present problem and the solution to the past problem most similar to the present one is then adopted.
2. Case-based reasoning lends itself to analogical reasoning. The use of analogies with past cases is how it solves new problems. It is intuitive to both developers and users.
 3. The knowledge acquisition process is considerably simplified since the case library may already exist as a corporate documentation, possibly even in an electronic database.
 4. The base of experience used can be that of an entire organization, rather than that of a single individual.

An example of using case-based reasoning comes from Compaq Computer of Houston, Texas, a company that operates in a highly competitive, customer service-oriented business environment. To improve customer service while reining in costs, Compaq began giving away case-based reasoning software to customers purchasing their Pagemarq printer.

The software knowledge base is a series of several hundred actual cases of Pagemarq printer problems. With this system running on their computer, owners no longer need to call Compaq's service department. Instead they run the software and describe the problem to the software. The system searches actual cases, selecting related ones. If necessary to further narrow the search results, the software will ask the user for more information. In the end, one or more cases relevant to the specific problem are displayed, along with their solutions. Now, customers can solve most of their own problems quickly without even a telephone call while Compaq saves \$10 million to \$20 million annually in customer support costs.

One of the challenges facing the developers of case-based reasoning software is to develop an efficient search algorithm for sifting through a myriad of historic cases. It is crucial that search time does not increase rapidly as the number of cases increases. A good search procedure should allow for multiple entries into a particular case through the use of pointers or links. In this manner problems that have been decomposed into smaller subsets can take advantage of the unique structure of the database and each of the subsets can then return information from entire cases. 

References

1. Allen, B.P. "Case Based Reasoning: Business Applications,," *Communications of the ACM*, March 1994, p 40-42.
2. Cook, R.L. "Case-based Reasoning Systems in Purchasing: Applications and Development," *International Journal of Purchasing and Materials Management*, winter 1997, Vol. 33, No. 1, p 32-39.
3. Gupta, U.G. "How Case-based Reasoning Solves New Problems," *Interfaces*, November-December 1994, Vol. 24, No. 6, p 110-120.
4. Kesh, S. "Case Based Reasoning," *Journal of Systems Management*, July-August 1995, Vol. 46, No. 4, p 14-21.
5. Li, H. "Case-based Reasoning for Intelligent Support of Construction Negotiation," *Information & Management*, August 1996, Vol. 30, No. 5, p 231-239.
6. Mehdi, O. "Case Discovery in Case-based Reasoning Systems," *Information Systems Management*, winter 1998, Vol. 5, No. 1, p 74-79.
7. Simon, H.A. *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations*, 1997, Fourth Edition, New York: The Free Press, 368 pp.