
Decision Support Systems: Directed and Nondirected Change

Mark S. Silver

*Anderson Graduate School of Management
University of California at Los Angeles
Los Angeles, California 90024*

The Decision Support Systems (DSS) literature is in general agreement that use of DSS leads to individual and organizational change, but there is no consensus as to whether DSS and their designers serve as agents for directed or nondirected change. Researchers have proceeded from two different sets of premises, drawing different conclusions about the nature of DSS. This paper considers both views, examining how differences in designers' attitudes toward change agency ought to be manifest in the features of the DSS they implement. Two attributes of DSS, "system restrictiveness" and "decisional guidance," are discussed as the basis for understanding differences in DSS following from differences in designer attitudes toward change. Using these two attributes, four DSS strategies for directed change and five strategies for nondirected change are presented.

Decision support systems—Change agency—System restrictiveness—Decisional guidance

One finds general agreement in the Decision Support Systems (DSS) literature that Decision Support Systems cause change in the processes through which decisions are made. Moreover, causing such individual and organizational change is generally seen as a proper function of a DSS. No consensus exists, however, concerning what the appropriate role of the change agent¹ should be.

Change agency comes in two varieties, reflecting two different attitudes on the part of the DSS designer. On the one hand, when designers comprehend that change will occur and deliberately attempt through a DSS to force the direction of that change, we have an instance of "directed" change. On the other hand, when designers understand that change will occur but do not try to influence the direction of that change, allowing it to be determined instead by the decision maker through DSS use over time, we have a case of "nondirected" change. By

¹A large body of research literature discusses the role of human change agents in affecting the behavior of individuals, groups, and organizations. This paper discusses a different kind of change agent: a computer-based system whose introduction into a decision-making environment causes change by intervening in the processes through which decisions are made.

default, when designers do not recognize their roles as change agents, a nondirected change situation follows.

Each of these two views of change agency—directed and nondirected change—is advanced and supported by prominent elements of the DSS research literature. The coexistence of the two positions raises numerous questions that are of importance to practitioners and should be of interest to researchers. Consider the following:

- Which view of change agency should be adopted? While some universally embrace one philosophy or the other, a broader perspective of DSS recognizes the legitimacy of both positions, with the choice between them situation specific. Which views apply in which situations?
- To what extent do the various prescriptions for DSS development depend upon the view of change agency one adopts? Are different development procedures appropriate in different change-agency situations?
- How should one's view of change agency in a given situation be reflected in the features of the DSS he or she produces? Should there be systematic differences between the features of DSS implementing directed change and those implementing nondirected change?

Surprisingly, scarcely any attention has been paid to these issues in the DSS research literature. In fact, the literature hardly notes that conflicting views exist. Papers presenting one or the other of the two positions do not even acknowledge the presence of an alternative perspective on DSS and change agency. Stabell (1983) stands alone in explicitly identifying the conflict, and even he devotes only a few paragraphs to its discussion. Boynton and Victor (1989) observe a related but different dichotomy in the literature on information search support. Moore and Chang (1983) address a closely related topic in their discussion of “meta-design considerations” for DSS, but they do not comment on the underlying tension in the literature.

This paper serves two purposes. First, it examines the treatment change agency has received in the DSS literature to date. Then, it considers the third set of questions just raised, how the features of a given DSS ought to reflect the underlying philosophy of change upon which the project is based. In order to address this issue, two DSS attributes—“system restrictiveness” and “decisional guidance”—are discussed. Using these two attributes, four DSS strategies for directed change and five strategies for nondirected change are presented.

Change Agency and the DSS Literature

The concept of change agency has long been recognized as important for understanding Decision Support Systems. Ginzberg (1978) found in a study of 29 systems that DSS require a substantially greater degree of individual change to be successful than do other types of information systems. Alter (1980), in his study of 56 DSS, noted that “implementation of a decision support system always constitutes some kind of change in a work environment” (p. 143). Keen and Scott Morton (1978) argue that the DSS designer must adopt the role of a “clinical, facilitative change agent.” Barki and Huff (1985) found that users more extensively used “those DSS that bring change than DSS that do not result in substantial changes” (p. 261).

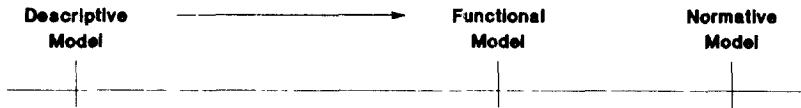


FIGURE 1. Schematic Interpretation of Gerrity's Decision-Centered Design Approach.

Note. This figure is an interpretation of the approaches presented in Gerrity (1970, 1971), Keen and Scott Morton (1978), and Stabell (1983).

The studies explicitly addressing the use of DSS and change agency have concentrated on the connections among *implementation processes*, system success, and organizational/individual change. Moreover, they have not compared different views of change agency such as directed versus nondirected change. In contrast, this paper focuses upon the relationship between *design features* of DSS and change agency. In particular, it considers how different views of change should lead to systems with different characteristics.

Either explicitly or implicitly, much of the current DSS literature rests upon the notion of DSS as agents for change. Which attitude—directed or nondirected change—characterizes this literature? Both. Even among the frequently cited components of the literature, both viewpoints are prominent. The terms “directed” and “nondirected” change do not appear often, but the underlying concepts do.

Directed Change

In one of the early DSS research efforts, Gerrity (1970, 1971) introduced an approach to DSS design and implementation that was subsequently endorsed by Keen and Scott Morton (1978) and expanded upon by Stabell (1978, 1983). The key components of this approach were normative, descriptive, and functional modeling, as illustrated in Figure 1. Under this scheme, the designer develops a normative model of how the decision “should” be made and a descriptive model of how the decision is currently being made. The functional model specifying the DSS is then defined as some point between the descriptive and normative models, intended to move the decision maker in the desired direction. The underlying attitude is clearly one of directed change.

Gerrity and the Portfolio Management System (PMS) he created provide a classic example of a designer viewing his role as that of an agent for directed change. Normative portfolio theory requires that a decision maker evaluate portfolios as a whole, trading off risk and return for the entire portfolio. Gerrity observed that account executives were violating prescriptive theory by evaluating individual securities and searching for appropriate portfolios, rather than examining portfolios and searching for appropriate securities. An important component of his system design, therefore, was to provide portfolio-oriented operators, thus encouraging the managers to move from security-oriented to portfolio-oriented analysis.

Nondirected Change

Sprague and Carlson (1982) identify six performance objectives for Decision Support Systems. The fifth item on their list is the following:

To support a variety of decision-making processes but not be dependent on any one. In other words, to provide support that is process independent and under full control of the user. (p. 95)

This assertion forms the foundation for a view of DSS as agents for nondirected change, with the user assuming responsibility for controlling the direction of any changes that occur. In essence, the DSS builder provides information-processing capabilities that are potentially valuable for performing the task, and the decision maker decides if and how to make use of these capabilities. Contrast this process-independent DSS philosophy with Gerrity's approach, where the system is intended to move the decision maker in a given direction.

Sprague and Carlson point to the Geodata Analysis and Display System (GADS) as a DSS that implements their philosophy. GADS provided its users with capabilities to access, display, and analyze data in making geographically-oriented decisions such as allocating policemen to beats. The purpose of the system was to improve the effectiveness of professionals by giving them access to useful, graphically-oriented, computer-based facilities, but not to impose any particular decision-making process on them. Studies indicated that processes changed and decisions improved with the use of GADS, while individual decision makers varied in terms of how they used the system to arrive at decisions.

Other well-known DSS development approaches can also be seen as reflecting a nondirected view of change agency. For instance, Stabell (1983) makes the following collective observation concerning evolutionary, middle-out, and adaptive design as presented by Moore and Chang (1983), Hurst, Ness, Gambino, and Johnson (1983), and Keen and Gambino (1983), respectively:

They say little about the direction or content of the changes to be achieved. The guidelines presented are consistent with the view of the DSS builder as an agent for nondirected change. It is the user who is responsible for defining the content of the change. The builder is merely a facilitator for user-directed change. (p. 225)

A Broader View of Change

Some DSS researchers have moved toward less extreme positions on change agency. The first step in this direction is to recognize the existence of both the directed and nondirected philosophies of change. Rather than dogmatically adopting a single perspective on change agency, this more enlightened viewpoint accepts the legitimacy of each approach and sees the choice between them depending upon the specific DSS project.

A recent paper by Boynton and Victor (1989) exemplifies this point of view. In the context of information-search behavior, they observe that the literature is divided between advocating systems that conform with managers' stated preferences and systems that allow managers the flexibility to determine their own search behavior. Acknowledging that in some situations these approaches may be appropriate, they argue that in other situations a completely different strategy may be more desirable: designing an information system that directs or controls users' search behavior. Such controls may be needed, for instance, to avoid a mismatch between the search propensities of a manager and the search requirements of his or her task and organizational role. Empirical research is in progress to understand better when directed and when nondirected search is appropriate.

A second step from the extreme positions not only recognizes the legitimacy of each philosophy of change, but sees a given DSS project as an opportunity for combining the two philosophies. That is, a system may be designed to direct some

decision-making changes and to allow for other, unplanned changes, as well. From this mixed viewpoint, the fundamental design issue is not choosing between directed and nondirected change, but deciding how much of each underlying philosophy should be reflected in the system.

Keen and Gambino's (1983) prescriptions for DSS design illustrate this second step. Although Stabell has tagged their view with the nondirected-change label, in fairness, they appear to advocate a mixture of the two agency views. In particular, their approach incorporates the descriptive and prescriptive mappings suggested by Gerrity. When described elsewhere (Keen 1980), the adaptive design approach again appears to combine both points of view.

Stabell has also branded Moore and Chang's (1983) "evolutionary approach" as nondirected change. In the same article, however, Moore and Chang characterize the nature of DSS interventions by defining a continuum of designs, with "strong" designs at one extreme and "weak" designs at the other. A "strong" design deliberately attempts "to shape or refine the user's decision-making process," while a "weak" design follows "the user's current preferences and existing capabilities" (p. 174). The "strong" and "weak" endpoints map nicely into directed and nondirected change, respectively. The interior points must then represent mixed approaches.

This paper takes only the first step in the direction of a broader view of DSS and change agency, studying closely—but independently—the cases of directed change and nondirected change. Taking the second step, a combined approach, would be premature at this time. While strategies combining directed and nondirected change constitute a topic worthy of research in the future, we must have a much better understanding of the two extreme cases before we will be in a position to consider points between them.

Attributes of Decision Support Systems

If we are to study how different change-agency views should affect DSS characteristics, we require a mechanism for describing and comparing the features of Decision Support Systems. Silver (1988c) has proposed a three-tiered approach to describing and differentiating Decision Support Systems, with each successive tier constructed upon the descriptive information of the preceding levels. The first tier is a catalog of the system's functional capabilities—that is, a simple statement of what the system can do. The second tier builds upon the first, describing how these functional capabilities are packaged. This tier portrays how the system appears to its users. The ROMC approach (Sprague and Carlson 1982; Carlson 1983), consisting of representations, operations, memory aids, and control devices, represents a popular way of performing this level of analysis. The third tier—referred to as the system attribute level—addresses fundamental properties of the DSS as a whole that determine its likely effect on users' decision-making processes. Rather than focussing on what the *system* can do, the attributes concentrate upon what *users* can and will do with the system.

Any significant differences stemming from opposing philosophies of change agency should be manifest in the effects systems have on their users' decision-making behavior. Our attention, therefore, focuses on the system attribute level. Two attributes—"system restrictiveness" and "decisional guidance"—are studied here.

In preparation for examining these system attributes, it is useful to distinguish two aspects of decision making: *structuring* the decision-making process and *executing* the decision-making process. Structuring the process involves selecting a problem representation and then defining the macro process, the ordered set of information-processing and problem-solving activities to be performed. Structuring the process can be seen as a “meta-choice” problem of deciding how to decide. Executing the process entails actually performing the various information-processing and problem-solving activities.

For example, in a multi-attribute problem such as buying a car, renting an apartment, or choosing a college, structuring the process might be accomplished by deciding to use a conjunctive elimination rule to reduce the set of alternatives and then to employ a scoring method to select a winner from the reduced set. Once this meta-decision is made, the process is executed and the decision made by actually performing the elimination and running the scoring model.

If we are to understand how a DSS affects decision-making behavior, we need to consider if and how the DSS affects structuring the process as well as if and how it affects executing the process. The restrictiveness and guidance attributes each address both process structuring and process execution.

*System Restrictiveness*²

It may seem counterintuitive to speak of the restrictiveness of a Decision Support System. After all, when a manager receives a DSS, his or her information-processing capabilities are expanded, not reduced. Nonetheless, a DSS can be at once expansive and restrictive. Since any given DSS will include some finite set of functional capabilities, when a decision maker relies on a given system to solve a problem, his or her decision-making process is constrained by that system’s functionality.

DSS are frequently implemented in situations where unique, well-defined decision-making processes do not exist. In situations such as these, where defining an appropriate decision-making process is itself a major concern, which features are included in a DSS and which are excluded play a critical role in determining the process that is ultimately followed. Nonetheless, when analyzing Decision Support Systems, we tend to focus on the question “What features are to be built into the system?” and disregard the question “What features are not to be included?” If our purpose is simply to describe what the system can do, then answering only the first question may be adequate. If we wish to understand how the system affects decision-making processes, however, both questions are equally important.

System restrictiveness is formally defined as follows:

System Restrictiveness: the degree to which and the manner in which a Decision Support System limits its users’ decision-making processes to a subset of all possible processes.

System restrictiveness is portrayed schematically in Figure 2. The outer ellipse corresponds to the universe of decision-making processes for solving a given problem, while the inner ellipse represents those processes supported by a particu-

²This section closely follows and summarizes the presentation in Silver (1988b).

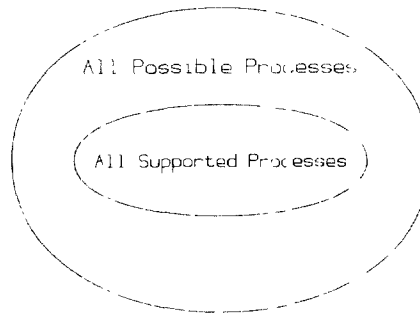


FIGURE 2. System Restrictiveness.

lar DSS. The *degree* of restrictiveness is determined by the relative sizes of the outer and inner ellipses, whereas the *manner* of restrictiveness is determined more specifically by what is inside and what remains outside the inner ellipse. The manner of restrictiveness is often of greater interest than the degree of restrictiveness. That is, we are usually more concerned with *how* the DSS is restrictive than with how *restrictive* the DSS is.

How DSS Restrict. DSS can restrict the structure as well as the execution of decision-making processes. The most obvious way of restricting the structure of a decision-making process is to exclude from the system support for certain information-processing activities. For example, DSS can prevent users from employing multi-attribute utility models by excluding the necessary operators. Conversely, DSS can force users to employ the multi-attribute utility approach by providing only operators that support this one choice rule. Similarly, DSS can be designed to include or exclude operators supporting a variety of other processes such as elimination by aspects (Tversky 1972) and analytic hierarchies (Saaty 1977).

Another manner of restricting the structure of decision-making processes is by constraining the order in which operators can be invoked. For instance, after projecting a budget with a deficit, users may be required to balance the projected budget before invoking any other operators. Before running a multiple regression, users may be required to check the Durbin-Watson statistic to test for autocorrelation in the data. Immediately after solving a linear program, users may be required to perform sensitivity analysis. In each of these examples, users are not only constrained by the set of capabilities included in the system, but also by strong sequencing rules that constrain the order of information-processing activities.

DSS can also restrict the execution of decision-making processes. By limiting user control over the parameters and other inputs to its operators, a DSS can constrain users' opportunities to perform evaluative and predictive judgments. For instance, a multi-attribute utility model might restrict the weight users can place on different attributes. A linear program might restrict users to predefined values for objective function coefficients, constraint coefficients, and bounds for constraints, rather than allowing users to determine these values for themselves. A stepwise regression operator might force users to accept a predefined significance level as opposed to allowing them to choose their own levels of significance.

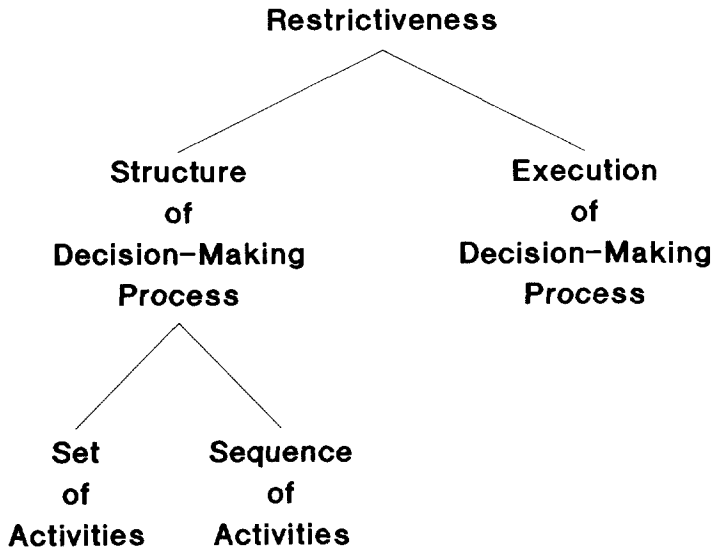


FIGURE 3. How DSS Restrict Decision-Making Processes.

Of course, how a DSS restricts its users' decision-making behavior is not exclusively a product of the system's features. Experimental results (Silver 1988a) show that perceptions of system restrictiveness differ from one user to another. In terms of Figure 1, what matters is not the full set of possible processes for solving a problem, but those processes that a given user sees as candidate solution techniques. Similarly, what matters is not the full set of capabilities provided by a DSS, but those that a given user is willing and able to operate effectively. How decision-making processes are restricted, therefore, is a function of the interaction between system and user.

The tree in Figure 3 summarizes the ways DSS contribute to restrictions in their users' decision-making processes. Since the decision-making process is the route through which decisions are reached, each of these forms of restriction can have major impacts upon the ultimate decision outcomes.

Determining Restrictiveness. A number of design objectives play a role in determining how restrictive a given DSS should be. While most of these objectives pertain to the effects the DSS will have on decision-making behavior, one nondecisional objective supersedes all others in importance: implementing a system that is used! A Decision Support System cannot successfully achieve its objectives, if it is never used. Bennett (1983) observed that DSS users are often discretionary users, who may eschew systems they find not to their liking. Consequently, a system's restrictiveness must be such that it promotes, rather than inhibits, use of the system.

Controlling system restrictiveness to promote DSS use is a double-edged sword, since either too much or too little restrictiveness may discourage decision makers from using the system. Moreover, how much restrictiveness constitutes too much or too little is situation specific.

Consider, first, too much restrictiveness. If a decision maker has a preferred approach to solving a particular problem, and if that approach is not supported by the DSS, he or she may simply choose not to use the system. Moreover, since flexibility is often considered an important feature of DSS, users who “feel” generally too constrained by a DSS may avoid using it. For these reasons, designers need to avoid making a system overly restrictive.

Now consider a DSS with too little restrictiveness. Such a system may overwhelm its users, presenting so many different capabilities and options that they are unable to choose among or operate the system’s features effectively. Indeed, Alter (1981) observes that as systems become more flexible they “often become more *difficult* to use, especially by less experienced users” (p. 15). A minimally restrictive DSS may appear, therefore, to be very difficult to use; decision makers may either not even try or very quickly stop trying to use it. To ensure that their intended users are both willing and able to use the DSS, therefore, designers may choose to make a system more restrictive than they might otherwise.

The remaining objectives bearing on system restrictiveness concern how a DSS is used and, in particular, how that DSS is likely to affect the processes through which decisions are reached. Let us start with the objectives favoring greater restrictiveness. Many of these objectives are found in situations where the “client” (Moore and Chang 1983) who commissions a DSS is not the same as its users.

- **Prescription.** Restrictive DSS can be used to prescribe normative decision-making techniques such as multi-attribute utility theory or normative portfolio theory. Restrictive DSS can also be used by managers to impose the techniques they deem appropriate on their subordinates. For instance, Boynton and Victor (1989) propose that, in some situations, an information system can serve as a lever for managers to control the information-search behavior of their subordinates.

Alter (1980) observed that sharing a DSS can promote communication and consistency of decision making within an organization. Restrictive DSS that prescribe common decision-making techniques, therefore, should be especially useful in promoting consistency, coordination, or communication among individuals or organizational units.

- **Proscription.** Restrictive DSS can be used to prevent users from following particular decision-making processes. A restrictive DSS might proscribe processes that manifest undesirable characteristics such as systematic cognitive biases. For example, to reduce a “confirmation bias,” Jacobs and Keim (1988) experimented with a knowledge-based DSS that limits user discretion in data retrieval.

- **Providing Structure to the Decision-Making Process.** Even when a DSS is not being used for prescription or proscription, building a restrictive system may be desirable to add structure to the decision-making process. Users might otherwise be overwhelmed by a system’s many options and be unable to use it effectively. Carroll and Carrithers’s (1984) experiment using “training wheels” to constrain users and facilitate ease of learning represents a comparable approach in the context of word processing packages.

- **Fostering Structured Learning.** In his study of 56 DSS, Alter (1980) found that learning was frequently a by-product of DSS use, but that sometimes DSS were deliberately designed with the goal of fostering learning. Restrictive systems can help meet this objective by providing structured learning. For instance, restrictive DSS can train managers in a given decision-making technique by marching them

through the steps they must follow. Restrictive DSS can take users on a “guided tour” of a database, systematically exposing them to various pieces of information.

Now, consider the objectives favoring lesser restrictiveness. Many of these can be seen as special cases of Sprague’s (1980) managerial objective that DSS be process independent and user controlled. Moreover, most of Keen’s (1980) arguments for adaptive DSS design can also be used to make a case for lesser restrictiveness.

- **Providing a Broad Repertoire of Decision Support Tools.** Keen (1980) notes that DSS designers and users are often unable to provide adequate functional specifications for DSS. Indeed, given the unstructured nature of the tasks being supported, one approach to designing DSS is to provide a broad collection of information-processing capabilities from which decision makers can choose as needed. This approach is consistent with Huber (1983), who argues that “DSS designs should enable their users to employ a variety of approaches to their decision tasks” (p. 575). Pursuing this objective leads naturally to building less restrictive systems.

- **Supporting Changing or Multiple Decision-Making Environments.** Sprague and Carlson (1982) note that “[i]t is widely recognized that DSS must be flexible because the environment, the tasks, and the users of DSS are subject to frequent change” (p. 130). One way of achieving such flexibility is through minimally restrictive DSS.

DSS are often intended to support multiple decision makers or multiple decision-making environments. These objectives, too, favor less restrictiveness. For instance, Carlson (1983) argues that a DSS should not enforce or capture a particular decision-making pattern if it is to support several decision makers. Hogue and Watson’s (1984) study of DSS in eighteen organizations found that because the DSS tended “to support multiple users, they were designed to be flexible” (p. 122).

- **Promoting Creativity.** Support for idea processing and creativity is often seen as an important objective for a DSS (Young 1985; Elam, Henderson, Keen, Konsynski, Meador, and Ness 1985; Elam and Mead 1987). One means of facilitating creativity is by providing users with a free hand—that is, a nonrestrictive DSS—in studying the problem under consideration.

- **Fostering Exploratory Learning.** Keen (1980), Alter (1980), and Hurst, Ness, Gambino, and Johnson (1983), among others, claim that individual or organizational learning is an important benefit following from the use of DSS. By experimenting with a nonrestrictive DSS and exploring its capabilities, decision makers can learn much about the nature of their decision-making problem.

The objectives favoring greater and lesser restrictiveness are summarized in Table 1.

Change Agency. Clearly, there is some connection between change agency and system restrictiveness. At first glance, one might conclude that the two concepts coincide, that directed change corresponds with constructing restrictive systems and nondirected change with building nonrestrictive systems. Although DSS built from a philosophy of directed change may tend to be more restrictive than those built from the opposing viewpoint, the relationship is not so simple. For each philosophy of change, designers must trade off the various specific objectives in a given situation to determine exactly where along the restrictiveness continuum the

TABLE 1
Objectives Affecting System Restrictiveness

Objectives Favoring Greater Restrictiveness	Objectives Favoring Lesser Restrictiveness
<ul style="list-style-type: none"> • Promoting Use • Prescription <ul style="list-style-type: none"> -Of a Normative Approach -For Coordination/Consistency • Proscription • Providing Structure to the Process • Fostering Structured Learning 	<ul style="list-style-type: none"> • Promoting Use • Providing a Broad Repertoire of Decision Support Tools • Supporting Multiple or Changing Decision-Making Environments • Promoting Creativity • Fostering Exploratory Learning

DSS should be placed. Moreover, the picture is not yet complete. A second system attribute, guidance, must be introduced before we can understand fully the relationship between change-agency philosophies and DSS features.

Decisional Guidance

In the course of operating a Decision Support System, users typically encounter a variety of opportunities for exercising judgment and making choices. When users are given a significant amount of discretionary power, they may require guidance to help them take advantage of and cope with the many options available to them. The restrictiveness attribute tells us how much discretion a DSS allows decision makers. To understand how a DSS is likely to affect decisional behavior, however, we also need to know if and how that DSS aids its users in exercising the freedom they are given.

Stabell (1983), following on Meador and Ness (1974), distinguishes between passive and active understanding of a system, where "passive understanding refers to the mechanics of system use," and "active understanding refers to how to use the system in the task at hand" (p. 224). Analogously, one can distinguish between "mechanical" and "decisional" guidance. Mechanical guidance helps users with the mechanics of operating a system's features, often a matter of knowing when to push which buttons. By far the most prevalent type of guidance today, it is provided mainly in the form of look-ahead menus, context-sensitive help screens, and status lines. In contrast, decisional guidance helps users deal with the decision-making concepts involved in choosing among and interacting with a system's information-processing capabilities. Although more rare at the present time, this is the type of guidance that is of primary interest in understanding how DSS affect decision-making behavior.

Decisional guidance has been defined as follows (Silver 1986, 1988c):

Decisional Guidance: the degree to which and the manner in which a Decision Support System guides its users in constructing and executing decision-making processes, by assisting them in choosing and using its operators.

Users may benefit if a DSS that supports many different solution techniques would also provide decisional guidance for the "meta-process" of structuring the decision-making process. Indeed, constructing a decision-making process—decid-

ing how to decide, as it were—is frequently the most difficult part of solving a problem, posing many tough questions for decision makers. For instance, given a particular multi-attribute problem, should a process method or a scoring method be used to choose among alternatives? Should a compensatory or noncompensatory rule be used? Should a serial approach be used, where solutions are generated and evaluated one at a time, or should a parallel process be employed, where a set of alternatives is generated that compete one with another. Decisional guidance assists users in recognizing and confronting process issues such as these.

However difficult it may be, constructing the decision-making process is only one step toward reaching a decision. The chosen solution method must, of course, be executed. Here, too, decisional guidance might be valuable. The conventional wisdom concerning DSS is that they combine the best features of humans and machines; the human decision makers perform the judgmental tasks, and the machines perform other information-processing activities. Nonetheless, human decision makers can benefit from computer-based guidance at those points in the decision-making process where they must execute judgment.

Consider, for example, a decision maker employing an elimination by aspects approach to choosing a city in which to live. A simple DSS for implementing this choice rule might allow its users to enter attributes and acceptable ranges for their values, producing a list of those cities satisfying the specified criteria. While such a system does aid decision makers by performing the necessary database searches, it does not assist with the critical judgmental tasks of choosing and ordering attributes and defining acceptable ranges. Another system might provide such decisional guidance.

Forms of Guidance. Instances of decisional guidance can be classified in a number of different ways. The definition itself suggests one distinction: guidance for structuring processes (choosing and ordering operators) versus guidance for executing processes (using operators). Another distinction addresses the informational content of the guidance, differentiating suggestive guidance from informative guidance.

Suggestive guidance proposes courses of action to the user. For instance, in terms of structuring the decision-making process, a DSS might suggest which operator to invoke next. Similarly, during process execution, a DSS might recommend values to be used as inputs to various algorithms. Consider a few examples: Kobashi (1984) proposes a tables-oriented decision aid (TODA) for multi-attribute decision making that might incorporate “suggestion” to assist less experienced decision makers with the temporal structuring of the process. The “Spreadsheet Manager” described by Fjelstad and Konsynski (1986) guides users in the selection of appropriate spreadsheet models or templates. Wedley and Field (1984) describe a “predecision” support system that draws upon decision-making research³ as it interacts with the user to recommend decision styles, methods, and participants.

In contrast, informative guidance provides users with pertinent information without indicating how the user might proceed. For example, users might be given a list of available operators together with an analysis of how they differ with respect to their decisional properties. Similarly, users might be given tables of reference data to help them choose input values for operators. Brennan and Elam

³For instance, Vroom and Yetton (1973) and Stumpf, Zand, and Freedman (1979).

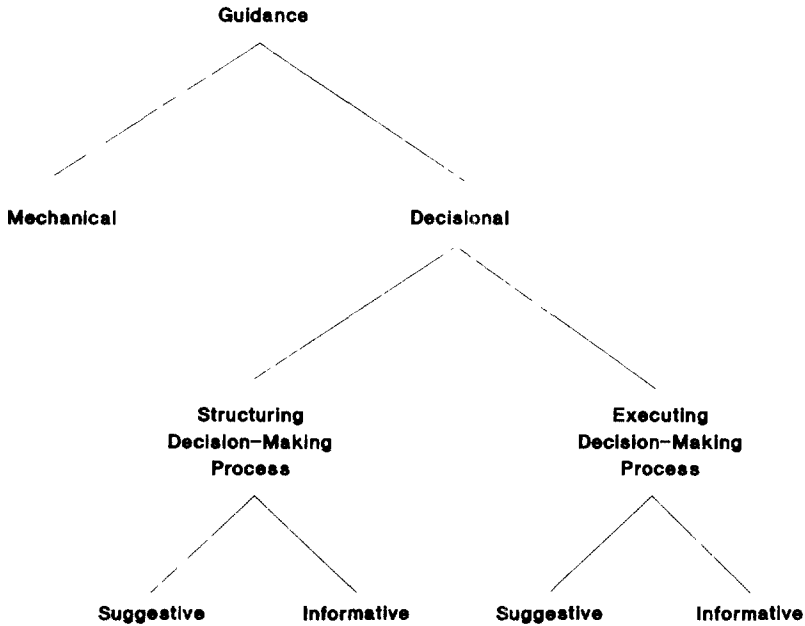


FIGURE 4. Forms of Guidance.

(1986) observe that guidance in a model-based DSS might address “what to do next” by providing “clues as to interesting or important changes to the model structure or parameters” (p. 136).

Figure 4 summarizes in hierarchical format the different types of guidance considered thus far. A given DSS can include instances of any or all of these forms of guidance. In addition to distinguishing the forms guidance can take, one can also distinguish the methods by which guidance is delivered: guidance can be provided to users automatically or only on request. The various types of guidance and delivery methods included in a DSS are likely to have significant effects on how the system is used and on the decisions that are made.

Providing Guidance. The first requirement for providing decisional guidance is *opportunity*. An interaction with system restrictiveness comes into play here, since highly restrictive DSS limit users’ discretion and hence provide relatively little opportunity for meaningful guidance. A DSS that is minimally restrictive, however, offers considerable opportunities for guiding decision makers in its operation. In general, as shown in Figure 5, the less restrictive a DSS, the greater the opportunity for providing decisional guidance.

Given opportunity, the second requirement is *motive*. There are two different motivations for including decisional guidance in a DSS. First, a designer may choose to include guidance in a DSS for the purpose of influencing decision makers. While allowing users the freedom to do as they please, the designer may nonetheless wish to encourage certain types of behavior. For instance, Elam and Mead (1987) suggest that a DSS whose purpose is to enhance creativity should provide feedback with “depth and positive tenor” to encourage prolonged alternative generation and delayed judgment.

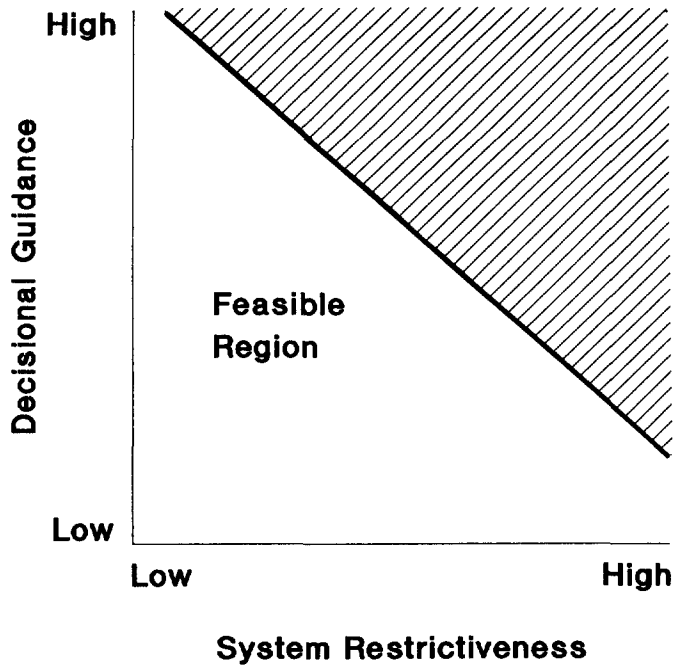


FIGURE 5. Feasible Combinations of Restrictiveness and Guidance.

A second motivation for providing guidance is simply to build a more supportive DSS by helping users navigate through its complexity, structure their decision-making processes, and exercise judgment. The designer does not advocate any particular decisional behavior, but tries only to support users as they decide for themselves how to proceed.

The two motives are closely related to the distinction between suggestive and informative guidance, but there is not a one-to-one relationship between these concepts. Designers who seek to influence decision makers will usually provide specific suggestions, but they may also do so by providing carefully selected informative guidance. Designers who seek to support but not influence decision makers may rely heavily on informative guidance, but may also offer some suggestive guidance.

Given opportunity and motive, the remaining requirement is having the *means* to provide guidance. Building decisional guidance into a Decision Support System is much more difficult than providing mechanical guidance. Ironically, those systems offering the greatest opportunity and motivation are often the ones for which presenting meaningful guidance is most problematic. In situations where DSS users are unclear how to proceed and need decisional guidance, DSS designers may be equally unclear how to guide them. Designers may find that incorporating many functional capabilities into a system is much easier than guiding decision makers concerning their use.

Consider what appears to be a perfect target for decisional guidance: a complex, minimally restrictive DSS that allows its users to follow many paths and to choose among numerous options in arriving at a decision. Now, at any given point in the decision-making process, the best paths to follow or options to choose are likely to

be dependent upon the actions previously taken by the user. Consequently, meaningful guidance will need to be highly context sensitive, taking into account how decision makers have arrived at a particular juncture in the decision-making process.

Finding a means to provide highly context-sensitive guidance presents a formidable challenge to system designers, who must anticipate the various contexts that can occur and make sure the system offers suggestions or information uniquely suited for each context. What can be the sources of such guidance? One possibility is for the designer to predefine the various guidance messages and build them into the system, an arduous task for large or complex systems.

Alternatively, the system can be designed to generate its own guidance messages dynamically by employing inferencing techniques from the field of artificial intelligence. For instance, a DSS could “learn” from its users’ behavior and make suggestions based on what was done previously in similar situations. Fjelstad and Konsynski (1986), Liang and Jones (1987), and Manheim (1988) have proposed DSS approaches that can be applied to this problem. Whatever approach is used, it will still require significant designer effort.

Once a means has been devised for creating meaningful guidance, a means must also be constructed for delivering that guidance to the user. In a complex system, ensuring that the context-sensitive messages are presented to the user at the correct point in time is itself a major task. Here, too, building intelligent mechanisms into the DSS could be advantageous.

Design Strategies

If we accept the facts that (1) there are two significantly different views of DSS and (2) DSS differ from one another in their features, then the natural question to ask is “What DSS features are logically consistent with which points of view?” The discussions of system restrictiveness and decisional guidance suggest that each of these attributes has a role to play in answering this question. Moreover, we must bear in mind the relationship between the two attributes: the greater the restrictiveness, the less opportunity for guidance.

Analyzing the system features consistent with each of the two philosophies—directed change and nondirected change—yields a separate set of design strategies, as shown in Table 2. The principal component of all these strategies is the judicious design of system restrictiveness and decisional guidance. Ideally, we would like a design strategy to be based entirely on system features such as restrictiveness and guidance. But by themselves, restrictiveness and guidance are not always sufficient to realize the desired type of change. Often another ingredient is necessary. Indeed, computer-based systems are complex interventions, consisting not only of the systems themselves but of organizational policies and procedures as well (Kling and Scacchi 1982). Therefore, while some of the strategies considered here are “pure” in that they consist only of choosing restrictiveness and guidance appropriately, others complement careful design of the system’s features with a set of implementation tactics, such as training and coaching, that are external to the system.

This paper focuses on *design strategies*—that is, on designing system features consonant with change-agency objectives. Of course, a complete *implementation strategy* for a DSS will include many other items, such as involving users in design

TABLE 2
Strategies for Directed and Nondirected Change

(a) Directed Change		(b) Nondirected Change	
Intuitive Strategy DC	Highly Restrictive DSS	Intuitive Strategy NDC	Minimally Restrictive DSS
Alternative Strategy DC-A	Highly Restrictive DSS with organizational mechanisms to promote use	Alternative Strategy NDC-A	Not-so-minimally Restrictive DSS
Alternative Strategy DC-B	Not-so-highly Restrictive DSS with Decisional Guidance	Alternative Strategy NDC-B	Minimally Restrictive DSS with Decisional Guidance
Alternative Strategy DC-C	Not-so-highly Restrictive DSS with training/coaching	Alternative Strategy NDC-C	Minimally Restrictive DSS with training/coaching
		Alternative Strategy NDC-D	Minimally Restrictive DSS with organizational mechanisms to promote use

or acquiring top management support. Important as these other items are, the discussion here concentrates on the features of the system itself and not on the full implementation strategy. Implementation tactics are referenced in the discussion of design strategies only as they are needed to complement the roles played by the restrictiveness and guidance attributes.

Directed Change

The various design objectives favoring greater restrictiveness—prescription, proscription, providing structure, and fostering structured learning—are very much in the spirit of directed change. Indeed, when a philosophy of directed change is adopted, a natural starting place may be to contemplate a highly restrictive Decision Support System, one that includes the desired processes, excludes undesirable processes, and minimizes the number of other processes that might divert users from the desired ones (Strategy DC in Table 2). In particular, the current decision-making process might be excluded to force a change in problem-solving methods. In some directed-change situations, simply implementing a highly restrictive DSS may be a successful strategy for achieving the design objectives. For example, in order to direct users to use multi-attribute utility theory, a plausible approach is to build a DSS whose capabilities are limited to multi-attribute models. Similarly, to force managers to consider more external information and less internal data in planning, the amount of internal data accessible through a

DSS might be constrained. Likewise, to prevent decision makers from following their current processes, functions central to those processes might be excluded from a DSS.

In other instances of the directed-change philosophy, however, one or more objectives favoring lesser restrictiveness may also be present. For example, a DSS may be intended both to cause specific changes in decision-making behavior and to foster creativity and exploratory learning. One objective particularly likely to favor lesser restrictiveness is system use, since there is a significant danger that a highly restrictive DSS will be eschewed by discretionary users. In situations where too much restrictiveness will inhibit DSS use or leave other objectives unsatisfied, an alternative strategy must be adopted.

If system use is the only problem, then a possible solution is to implement the highly restrictive DSS together with organizational mechanisms that promote its use (Alternative Strategy DC-A). A likely mechanism is a policy of mandatory use. Such a policy might be instituted by a manager who is implementing a DSS for the purpose of changing his or her subordinates' decision-making behavior. Alter (1980) warns us, however, that a potential pitfall associated with mandatory use is "half-hearted" use. Half-hearted use of a DSS might fail to lead to the desired changes in decision-making behavior. Other organizational mechanisms to promote use, such as pressure from peers or competitors, might therefore be considered.

A much different design strategy for directed change would be to reduce the restrictiveness of the system, an approach aimed at overcoming the problem of nonuse as well as satisfying other objectives that favor lesser restrictiveness. The danger here is that without the restrictiveness to enforce the desired changes in decision-making processes, these changes may never take place.

Gerrity's (1970, 1971) Portfolio Management System (PMS) offers a case in point. Recall that Gerrity's objective was to move account executives from a security-oriented approach toward the normative portfolio-oriented approach to decision making. PMS included security-oriented operators as well as the portfolio-oriented operators Gerrity hoped account executives would eventually use. Unfortunately, a follow-up study conducted by Stabell (1974) several years after Gerrity's study revealed that, although PMS was still being used, the desired changes in decision making had not been realized. Evidently, given the opportunity to use either the old or new approaches, account executives remained with their original decision-making methods.

The portfolio management example reinforces the concern that a reduced-restrictiveness strategy may lack the means to bring about desired changes in decision-making behavior. Two solutions are possible here, one involving only system features and the other requiring activities external to the system. First, we can now introduce decisional guidance into the picture, since the lessened restrictiveness provides the opportunity and the directed-change philosophy constitutes the motive (Alternative Strategy DC-B). Without actually forcing users to follow certain processes, as restrictiveness would, guidance—especially of the suggestive variety—might nudge users in the desired direction. Stabell (1983) has coined the term "decision channeling" to refer to the property of systems "that serves to both support and shift the decision process" (p. 251). We can view decision channeling

as a special case of decisional guidance, where the system suggests changes in decision-making behavior to its users. Further research is required to identify the means for providing this form of guidance.

Instead of compensating for reduced restrictiveness by building decisional guidance into a DSS, a second reduced-restrictiveness approach attempts to direct change by training or coaching decision makers (Alternative Strategy DC-C). Training would emphasize using the system in a manner consonant with the desired changes in decision-making processes. Ongoing coaching might be used instead of or in addition to training in order to foster the desired decision-making behavior. This approach may be necessary or desirable given the difficulty of building meaningful, context-sensitive decisional guidance into a DSS. Given current technology, a human trainer or coach may be better able to channel a user's decision-making behavior than can any features we embed in the system. A key drawback of employing human coaches rather than computer-based guidance, however, is that managers may use the DSS in the coaches' absence, with no mechanism present to direct system use.

Huber (1983) suggests a similar approach in the context of cognitive style research and DSS design. Noting that it might be dysfunctional to build DSS that are restricted to their users' cognitive styles and that it might be equally undesirable to build systems that only complement their users' styles, he envisions building more flexible DSS that incorporate both decisional aids consonant with and aids complementary to decision makers' styles. Users would then be free to choose appropriate aids as required by particular tasks. Through training and coaching, decision makers can be educated to choose the most appropriate aid in a given situation. This approach need not be limited to cognitive style considerations; in many cases, flexible DSS can be built that support users' decision-making processes as well as alternative processes, with desired behavioral changes fostered by training and coaching.

The four strategies associated with directed change are positioned with respect to their restrictiveness and guidance on the right half of Figure 6. When the intuitive strategy (DC) of simply building a highly restrictive DSS is not appropriate, an alternative is formed by modifying system features, introducing external influences, or both. In Alternative Strategy DC-A, restrictiveness and guidance are unchanged, but pressure to use the system is added. In contrast, Alternative Strategies DC-B and DC-C reduce system restrictiveness and compensate by adding decisional guidance and training/coaching, respectively. Two of the strategies (DC and DC-C) are pure, in the sense that they involve only the design of system features, whereas the other two strategies (DC-A and DC-B) involve both system features and implementation tactics.

Nondirected Change

The nondirected philosophy of change expects change in decision-making behavior to occur, but does not try to influence the direction of that change. Since users must be given freedom to choose their own decision-making processes, the most intuitive strategy is to implement a minimally restrictive DSS (Strategy NDC in Table 2). Presumably, such a DSS will include facilities to support the current

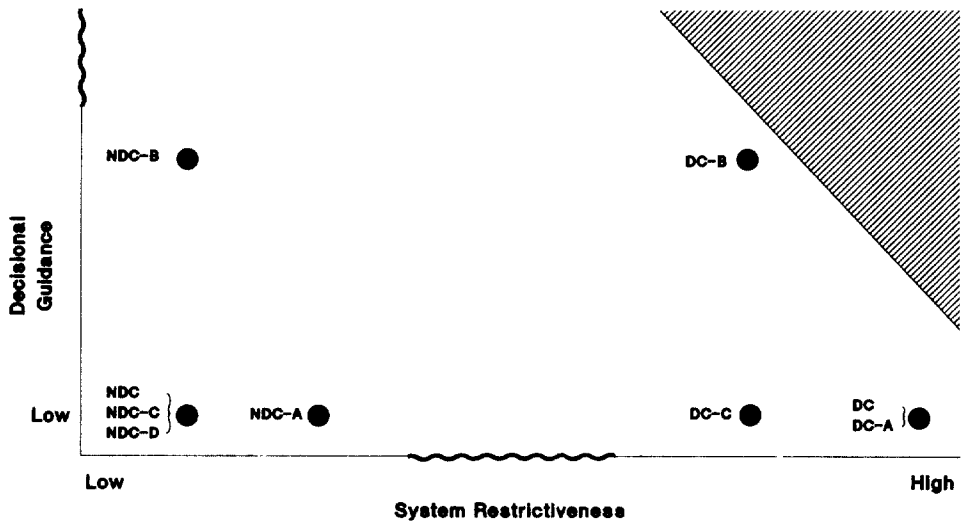


FIGURE 6. Strategies for Directed and Nondirected Change.

Note. Strategy DC-A adds organizational pressures for use, whereas DC-C adds coaching or training. Strategy NDC-C adds coaching or training, whereas Strategy NDC-D adds organizational pressures for use.

decision-making process as well as offer sufficient leeway for users to explore and experiment with other problem-solving approaches.

Just as the agent for directed change could not necessarily succeed with a maximally restrictive DSS, an agent for nondirected change might not wish or not be able to build one that is minimally restrictive. The designer's hope is that, without advocating specific decision-making changes, changes will nonetheless occur over time and that they will be for the better. Simply building a nonrestrictive DSS that supports the current process together with many others, however, may fail to accomplish this basic objective.

Stabell (1983) suggests that a DSS may lead to "further entrenchment of the existing, ineffective decision-making process that it supports" (p. 228). Minimally restrictive systems supporting existing processes seem especially vulnerable to this threat. Indeed, Moore and Chang (1983) note that unless the user is highly self-motivated, such systems may lead primarily to higher decision-making efficiency but not to greater decision-making effectiveness. Worse yet, we can imagine a very nonrestrictive DSS allowing users to move to a process inferior to their current process, "hanging themselves," as it were, with all of the rope the designer gives them. For instance, a minimally restrictive DSS allows more room for systematic cognitive biases and other decision-making weakness to creep into the process.

Another concern is that minimally restrictive DSS may prove to be very difficult for decision makers to use effectively. In providing users with a great deal of flexibility to define their own decision-making processes, those decision makers may be overwhelmed by the many options they confront, suffering from operator or information overload. Decision makers may also combine functional capabilities

in unproductive ways. Moreover, decision makers may find it difficult to retrace their steps and describe to others how decisions were reached.

These potential problems with nonrestrictive systems suggest that the simple strategy may need to be modified. One possibility is to increase somewhat the restrictiveness of the DSS in order to preclude entrenchment in the current process, proscribe potential decision-making errors, or reduce the overload of options suffered by users (Alternative Strategy NDC-A). In keeping with the philosophy of nondirected change, the increased restrictiveness serves not to impose a new decision-making process on users but to facilitate their effective use of the system as they determine for themselves how best to proceed.

Instead of increasing restrictiveness, another possible strategy is to seize the opportunity the minimal restrictiveness provides to add a significant amount of decisional guidance to the system (Alternative Strategy NDC-B). Unlike the “decision-channeling” guidance employed in the case of directed change, here the guidance is intended only to make the system more usable and to help users overcome the difficulties associated with nonrestrictive systems. The guidance tends to be more “informative” and less “suggestive,” although different specific motives will likely lead to different forms of guidance. Avoiding entrenchment in the current process and preventing decision-making blunders are motives that may entail suggestive guidance, while helping users cope with complex systems favors informative guidance.

Yet another possible strategy provides human trainers and coaches to counter the problems posed by nonrestrictive DSS (Alternative Strategy NDC-C). As in the case of directed change, training/coaching external to the system is seen as a substitute for decisional guidance embedded in the system. But just as the nature of guidance for nondirected change differed from that for directed change, so too does the nature of training and coaching. Here, the role of trainers and coaches is simply to enable users to appreciate the system’s capabilities and use them effectively, not to indoctrinate decision makers concerning the proper way to make decisions with the system.

Minimally restrictive DSS may create another problem. Recall that, with respect to system use, restrictiveness is a double-edged sword: either too much or too little restrictiveness can lead to nonuse. There is a danger, therefore, that the minimally restrictive DSS will not be used. The three modifications just proposed—increased restrictiveness, the addition of decisional guidance, or the provision of training/coaching—could potentially solve this problem. Alternatively, if nonuse is the only concern, then the minimally restrictive system can be augmented by organizational pressures, as already described for directed change, to encourage or require system use (Alternative Strategy NDC-D). Caution must be observed, however, since this approach addresses only the problem of nonuse and not the underlying problem of overloading the decision maker with capabilities.

The left half of Figure 6 positions the strategies for nondirected change in terms of their restrictiveness and guidance. When the intuitive strategy (NDC) of building a minimally restrictive DSS is not appropriate, the system design features can be modified either by increasing the restrictiveness as in Alternative Strategy NDC-A or by adding decisional guidance as in Alternative Strategy NDC-B. Instead of modifying system features, another possibility is adding training or

coaching as in Alternative Strategy NDC-C. To promote use of the system, the intuitive strategy can be augmented by organizational pressures for its use, forming Alternative Strategy NDC-D. Of the five strategies, three (NDC, NDC-A, and NDC-B) are pure strategies while the other two (NDC-C and NDC-D) include implementation tactics.

Research Implications

The preceding section studied the DSS design features that are logically consistent with each view of change agency. This analysis identified nine design strategies, four for directed change and five for nondirected change. Individually and collectively, these strategies raise a variety of research questions, most behavioral but some technological.

The behavioral questions, all of which require empirical investigation, concentrate on evaluating the logical strategies just proposed and on examining the actual strategies employed by DSS builders to date. Answering these questions requires new examinations of the development and use of DSS, since most of the DSS cases reported in the literature provide insufficient information concerning the change-agency views of the builders, the features of the systems, and the effects of the systems on the decision-making behavior of users. These three items—change-agency views, system features, and effects on decision-making—are the keys to evaluating the various strategies and their performance.

One important set of behavioral questions concerns how successfully each strategy meets its design objectives. For instance, how much restrictiveness and how little restrictiveness inhibit use of the DSS? Does suggestive guidance succeed in channeling decision makers into following new processes? Is a policy of mandatory use effective for directed change, or does it lead to half-hearted use of the system and little change?

In addition to evaluating the various strategies individually, we also need to compare the strategies with one another. We need to ask questions concerning when each strategy is most appropriate. For instance, what are the relative merits of building decisional guidance into a system as compared with providing human trainers and coaches? Similarly, when should we use increased restrictiveness and when should we use informative guidance to avoid overloading users with too many options?

Some of these questions can be studied in the laboratory; others require an organizational setting. For example, we can design laboratory experiments that measure the success of different forms of decisional guidance at channeling users' decision-making processes. Similarly, we can create laboratory experiments that compare the effects of decisional guidance with training and coaching. In contrast, studying the effectiveness of strategies that employ organizational mechanisms to promote use is more appropriate in a field setting.

Another line of behavioral research studies the historical and current practice of DSS designers. How prevalent are each of the two views of change agency? For each view, what combinations of system restrictiveness and decisional guidance have designers employed? How successful have these strategies been? What have been the consequences of building DSS whose features are not logically consistent with the adopted view of change agency?

The technological questions concern how to use computer-based technology to construct DSS exhibiting particular features. For example, what role can artificial intelligence techniques play in providing decisional guidance? Likewise, how can we define context-sensitive sequencing rules in a DSS that restrict the order of information-processing activities effectively?

This paper began by raising three questions concerning the role of DSS in directed and nondirected change. One of these questions, how system features should reflect the intended type of change, was considered here. Another, how analysis and development procedures should reflect the intended type of change, merits comparable attention. But most of all, we need to study the question of which type of change agency—directed or nondirected—is most appropriate in a given situation.

Acknowledgements. I am indebted to my colleagues, M. Lynne Markus and E. Burton Swanson, for their helpful comments on earlier versions of this paper and to my research assistant, Sidne G. Ward, for her help with this project. I also appreciate the excellent suggestions made by the associate editor and reviewers.*

*Gerardine DeSanctis, Associate Editor. This paper was received on September 12, 1988 and has been with the author 4½ months for 1 revision.

References

- Alter, Steven L., *Decision Support Systems, Current Practice and Continuing Challenges*, Addison-Wesley Publishing Company, Reading, MA, 1980.
- _____, "Transforming DSS Jargon into Principles for DSS Success," in *DSS-81 Trans.*, Donovan Young and Peter G. W. Keen (Eds.), Atlanta, GA, June 8–10, 1981, 8–27.
- Barki, Henri and Sid L. Huff, "Change, Attitude to Change, and Decision Support System Success," *Information and Management*, 9, 5 (December 1985), 261–268.
- Bennett, John L., "Analysis and Design of the User Interface for Decision Support Systems," in John L. Bennett (Ed.), *Building Decision Support Systems*, Addison-Wesley Publishing Company, Reading, MA, 1983, 41–64.
- Boynton, Andrew C. and Bart Victor, "Information System Design and the Control Imperative: A Modest Proposal," Darden School Working Paper Series, 1989.
- Brennan, J. J. and Joyce Elam, "Enhanced Capabilities for Model-Based Decision Support Systems," in Ralph J. Sprague, Jr., and Hugh J. Watson (Eds.), *Decision Support Systems: Putting Theory into Practice*, Prentice-Hall, Englewood Cliffs, NJ, 1986, 130–137.
- Carlson, Eric D., "An Approach for Designing Decision Support Systems," in John L. Bennet (Ed.), *Building Decision Support Systems*, Addison-Wesley Publishing Company, Reading, MA, 1983, 15–39.
- Carroll, John M. and Caroline Carrithers, "Training Wheels in a User Interface," *Comm. ACM*, 27, 8 (August 1984), 800–806.
- Elam, Joyce J., John C. Henderson, Peter G. W. Keen, Benn Konsynski, C. Lawrence Meador, and David Ness, "A Vision for Decision Support Systems," unpublished manuscript, 1985.
- _____, and Melissa Mead, "Designing for Creativity: Considerations for DSS Design," *Information and Management*, 13, 5 (December 1987), 215–222.
- Fjelstad, Oystein D. and Benn R. Konsynski, "The Role of Cognitive Apportionment in Information Systems," *Proc. Seventh Internat. Conf. Information Systems*, San Diego, CA, December 15–17, 1986, 84–98.
- Gerrity, Thomas P., Jr., "The Design of Man-Machine Decision Systems," Ph.D. dissertation, Massachusetts Institute of Technology, 1970.
- _____, "Design of Man-Machine Decision Systems: An Application to Portfolio Management," *Sloan Management Rev.*, 12, 2 (Winter 1971), 59–75.

- Ginzberg, Michael J., "Redesign of Managerial Tasks: A Requisite for Successful Decision Support Systems," *MIS Quart.*, 2, 1 (March 1978), 39-52.
- Hogue, Jack T. and Hugh J. Watson, "Current Practices in the Development of Decision Support Systems," *Proc. Fifth Intern. Conf. Information Systems*, Tucson, AZ, November 28-30, 1984, 117-127.
- Huber, George P., "Cognitive Style as a Basis for MIS and DSS Designs: Much Ado about Nothing?" *Management Sci.*, 29, 5 (May 1983), 567-579.
- Hurst, E. Gerald, Jr., David N. Ness, Thomas J. Gambino and Thomas H. Johnson, "Growing DSS: A Flexible, Evolutionary Approach," in John L. Bennett (Ed.), *Building Decision Support Systems*, Addison-Wesley Publishing Company, Reading, MA, 1983, 111-132.
- Jacobs, Sheila M. and Robert T. Keim, "An Experimental Study in Overcoming Hypothesis-Confirming Search Strategies in Computerized Information Retrieval Systems," *Proc. Ninth Internat Conf. Information Systems*, Minneapolis, MN, November 30-December 3, 1988, 81-89.
- Keen, Peter G. W., "Adaptive Design for Decision Support Systems," *Data Base*, 12, 1-2 (Fall 1980), 15-25.
- _____ and Thomas J. Gambino, "Building a Decision Support System: The Mythical Man-Month Revisited," in John L. Bennett (Ed.), *Building Decision Support Systems*, Addison-Wesley Publishing Company, Reading, MA, 1983, 133-172.
- _____ and Michael S. Scott Morton, *Decision Support Systems. An Organizational Perspective*, Addison-Wesley Publishing Company, Reading, MA, 1978.
- Kling, Rob and Walt Scacchi, "The Web of Computing: Computer Technology as Social Organization," *Adv. in Computers*, 21 (1982), 1-90.
- Kobashi, Yasuaki, "The Use of Suggestions in a Tables-Oriented Decision Aid," in Leif B. Methlie and Ralph H. Sprague, Jr., (Eds.), *Knowledge Representation for Decision Support Systems, Proc. IFIP WG 8.3 Working Conf.*, Elsevier Science Publishers B.V., North-Holland, Durham, U. K., July 24-26, 1984, 221-225.
- Liang, Ting-peng and Christopher V. Jones, "Design of a Self-evolving Decision Support System," *J. Management Information Systems*, 4, 1 (Summer 1987), 59-82.
- Manheim, Marvin L., "An Architecture for Active DSS," in Benn R. Konsynski (Ed.), *Proc. Twenty-First Annual Hawaii Internat Conf. System Sciences*, vol. III, 1988, 356-365.
- Meador, Charles L. and David N. Ness, "Decision Support Systems: An Application to Corporate Planning," *Sloan Management Rev.*, 15, 2 (Winter 1974), 51-68.
- Moore, Jeffrey H. and Michael G. Chang, "Meta-Design Considerations in Building DSS," in John L. Bennett (Ed.), *Building Decision Support Systems*, Addison-Wesley Publishing Company, Reading, MA, 1983, 173-204.
- Saaty, Thomas L., "A Scaling Method for Priorities in Hierarchical Structures," *J. Math. Psychology*, 15, 3 (June 1977), 234-281.
- Silver, Mark S., "Differential Analysis for Computer-Based Decision Support," Ph.D. dissertation, The Wharton School, University of Pennsylvania, Philadelphia, PA, 1986.
- _____, "User Perceptions of Decision Support System Restrictiveness: An Experiment," *J. Management Information Systems*, 5, 1 (Summer 1988(a)), 51-65.
- _____, "On the Restrictiveness of Decision Support Systems," in R. M. Lee, A. M. McCosh, and P. Migliarese (Eds.), *Organizational Decision Support Systems, Proc. IFIP WG 8.3 Working Conf.*, Elsevier Science Publishers B.V., North-Holland, Como, Italy, June 1988(b), 259-270.
- _____, "Descriptive Analysis for Computer-Based Decision Support," *Oper. Res.*, 36, 6 (November-December 1988(c)), 904-916.
- Sprague, Ralph H., Jr., "A Framework for the Development of Decision Support Systems," *MIS Quart.*, 4, 4 (December 1980), 1-26.
- _____ and Eric D. Carlson, *Building Effective Decision Support Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.
- Stabell, Charles B., "Design and Implementation of DSS: Some Implications of a Recent Study," Research Paper Series No. 252, Graduate School of Business, Stanford University, 1978.
- _____, "A Decision-Oriented Approach to Building DSS," in John L. Bennett (Ed.), *Building Decision Support Systems*, Addison-Wesley Publishing Company, Reading, MA, 1983, 221-260.
- Stumpf, Stephen A., Dale E. Zand, and Richard D. Freedman, "Designing Groups for Judgmental Decisions," *Acad. Management Rev.*, 4, 4 (October 1979), 589-600.

- Tversky, Amos, "Elimination by Aspects: A Theory of Choice," *Psychological Rev.*, 79, 4 (July 1972), 281–299.
- Vroom, Victor H. and Philip W. Yetton, *Leadership and Decision Making*, University of Pittsburgh Press, Pittsburgh, PA, 1973.
- Wedley, William C. and Richard H. G. Field, "A Predecision Support System," *Acad. Management Rev.*, 9, 4 (October 1984), 696–703.
- Young, Lawrence F., "Idea Processing Systems: Definitions, Concepts, and Initial Applications," in Joyce J. Elam (Ed.), *DSS-85 Trans.*, San Francisco, CA, April 1–4, 1985, 125–134.