

Beyond Spreadsheets: Tools for Building Decision Support Systems

The complexity and long development time inherent in building decision support systems has thus far prevented their wide use. A new class of tools, DSS generators, seeks to cut the lead time between development and deployment.

Hemant K. Bhargava
Carnegie Mellon
University

Suresh Sridhar
i2 Technologies

Craig Herrick
US Navy

Decision support systems (DSSs) are software products that help users apply analytical and scientific methods to decision making. They work by using models and algorithms from disciplines such as decision analysis, mathematical programming and optimization, stochastic modeling, simulation, and logic modeling. DSS products can execute, interpret, visualize, and interactively analyze these models over multiple scenarios.

In recent years, the growing popularity of online analytical processing, data warehousing, and supply chain management has led to an increased interest in the development of decision support systems. Decision support tools could assist decision-makers in problems involving risk management, the allocation of scarce resources, and the need to balance conflicting objectives. When well implemented and used wisely, DSSs can significantly improve the quality of an organization's decision making.

However, building a DSS requires significant expertise in decision analysis, programming, and user interface design. A DSS can also need to connect in real time with other enterprise applications, further complicating the task. For these reasons, the broad use of DSSs—compared with, say, database management systems—has not occurred.

What's missing is sophisticated tool support for developing DSSs. One current solution is a *DSS generator*,¹ a kit or environment for developing an application-specific DSS. DSS generators provide tools that make it eas-



ier and faster to develop models, data, and user interfaces that are customized to the application's requirements. Using a DSS generator reduces DSS development to a decision analysis task—which requires expertise in decision analysis and mathematical modeling—rather than a programming task. DSS generators are crucial to the success of DSSs in practice.

In this article, we describe the state of the art in DSS generator software, specifically in the realm of decision analysis methods (see the “Decision Analysis” sidebar).² Decision analysis techniques account for the uncertain, dynamic, and multicriteria aspects of decisions. Essentially, they aid the evaluation of alternatives in the face of trade-offs. Well-known decision analysis methods include decision trees and influence diagrams. Here we briefly describe the features of 11 commercially available DSS generators that specialize in decision

Decision Analysis

Decision analysis consists of theories, processes, and analytical methods for decision making that account for the uncertain, dynamic, and multicriteria aspects of decisions. Well-known decision analysis methods include decision trees, influence diagrams, multi-attribute utility models, and the analytic hierarchy process. Figure A shows samples of the first two, the most common methods employed in decision analysis. These methods allow decision-makers to evaluate decision alternatives in terms of the trade-offs between multiple decision criteria as well as the risk and uncertainties associated with each alternative.

A typical decision analysis problem that involves uncertainty is the *harvesting problem*. In this example, a farmer must decide when to harvest a crop of, say, tomatoes, to maximize his earn-

ings. An early harvest would realize the current crop but fetch a lower unit price. A late harvest of vine-ripened tomatoes is risky: It might fetch a higher unit price and even a higher yield, but some or all of the crop could be lost to frost.

The harvesting problem's structure and details can be modeled as an influence diagram or as a decision tree. We can transform most influence diagrams into decision trees. A decision tree representation, however, loses the conditional probability relationship between variables and can quickly become very big. Influence diagrams are compact and preserve logical relationships but are unsuitable for asymmetric problems—problems whose structure does not branch out symmetrically. Asymmetric problems comprise many useful decision analysis problems in the real world.

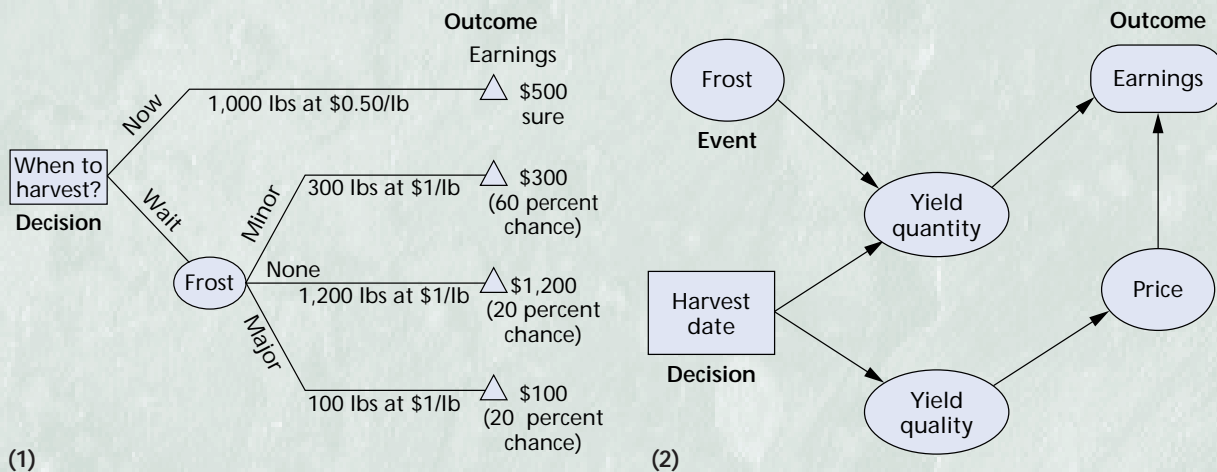


Figure A. Sample (1) decision tree and (2) influence diagrams, components of decision analysis.

analysis. Although not a comprehensive, complete analysis of these tools, this article clarifies the idea of DSS generators as DSS development environments and presents an overview of the progress in this area.

WHAT IS A DSS GENERATOR?

The word "generator" is somewhat of a misnomer. Although DSS generators are indeed used to develop DSS applications for specific decision problems, they do not behave like a typical code generator (which automatically generates programming code from a set of requirements specifications), or a software development environment. Programmers use a development environment to write and then compile source code. The resulting stand-alone, runtime object can be executed without the development environment.

Using a DSS generator, in contrast, is more akin to using an interpreted programming language—you must always have the interpreter to execute the code. As another example, consider a spreadsheet as an application development environment. Application developers use certain spreadsheet features to create

mathematical models, data structures, macros, input forms, and other objects. However, spreadsheets don't have compilers and don't produce runtime objects. Users must have the spreadsheet system to execute the application.

As an application development environment, a DSS generator is more like a spreadsheet than a C++ development environment. Developers work with a DSS generator, which provides certain generic functions (discussed later). DSS developers can use these functions to build models and data for the application, speeding up the DSS development process. However, DSS generators don't produce runtime objects, and the applications they generate can be executed only if the user has the generator program as well.

Therefore, a DSS generator is a code generator in only a limited sense: It provides a visual interface for model development and data definition. A DSS generator transforms the resulting visual objects into code in a lower level model-definition language. This lets DSS developers focus on decision analysis and model development rather than programming constructs—

data structures, functions, procedures—and the interaction among them.

Required functionality

So what does a DSS generator need to do? Its functionality depends on what we expect the DSS to do. A DSS should have the necessary set of models, data, and user interfaces for the application in question. It must include the necessary input and output representations and control mechanisms. Thus, a DSS generator should encompass

- a *modeling paradigm*, that is, specific methods for solving decision problems;
- *model and data definition features*, for defining model schemas and for entering data;
- *model and data management features*, including a specialized mechanism for maintaining, storing, retrieving, and executing models and for defining and manipulating data;
- *analytical methods and visual representations*, for exploring relationships between outcome variables and various controllable variables (including decision variables and user preferences) and/or uncontrollable variables that influence outcomes;
- *user interfaces*, either general-purpose and/or the capability to create a specialized user interface; and
- *model and data interchange features*, for communicating with external programs such as databases or external user interfaces.

Commercial products

DSS generators were primarily a theoretical concept in the 1980s, but several commercial products have emerged in the past few years. These products

provide a fairly comprehensive set of DSS features. We limit this discussion to products that apply decision analysis methods; an earlier survey³ provides a more complete list of systems.

These products let a developer build a DSS application simply by specifying the necessary models and data. They offer a visual and/or textual language for building model schemas, features for model solution and analysis, and commands and representations for visualizing model results. They minimize development effort by offering a generic (but usually inextensible) graphical user interface, generic (but again, inextensible) data management features, and generalized solution algorithms and analysis tools.

Using these products, someone who knows the relevant modeling paradigm and the problem domain can develop a DSS application in a few hours or days—a significantly shorter time than previously possible. Such DSS generators are now commercially available.

We evaluated the 11 DSS generators listed in Table 1, comparing the decision analysis methods they support and their strengths and limitations. All of them are based on one or more decision analysis methods.

EVALUATING DSS GENERATORS

When we discuss the capabilities, strengths, and limitations of specific products, we do so in an anecdotal sense. Our aim is to give readers concrete examples and a flavor for the features found in various systems. We organize the discussion according to DSS functionality.

Modeling paradigm

Decision analysis products employ two major categories of methods for modeling and solving decision problems.

Table 1. DSS generators considered for evaluation.

DSS generator	Vendor name	URL
Aliah Think!	Aliah Inc.	http://www.aliah.com
Analytica	Lumina Decision Systems Inc.	http://www.lumina.com
Criterion Decision Plus	InfoHarvest Inc.	http://www.infoharvest.com
Decide Right for Windows	Avantos Performance Systems Inc.	Not available
Decision Analysis by TreeAge (DATA)	TreeAge Software Inc.	http://www.treeage.com
Decision Pro	Vanguard Software	http://www.vanguardsw.com
Decision Programming Language (DPL)	Applied Decision Analysis	http://www.adainc.com
Expert Choice Professional	Expert Choice Inc.	http://www.expertchoice.com
Logical Decisions for Windows (LDW)	Logical Decisions	http://www.logicaldecisions.com
PrecisionTree	Palisade Corp.	http://www.palisade.com
Which & Why	Arlington Software Corp.	http://www.arlingsoft.com

The first category—*decision making under uncertainty*—includes methods for dealing with uncertainty about data values, user preferences, and other aspects about the decision problem, and for analyzing decision alternatives in terms of both return and risk. These methods include decision trees (employed by Decision Pro and PrecisionTree), influence diagrams (Analytica), or some proprietary variation of these. Decision Analysis by TreeAge (DATA) allows users to develop

a general influence diagram and convert it to a detailed decision tree. Decision Programming Language (DPL) provides users a way to represent problems as a hybrid of decision trees and influence diagrams.

The second category—*multicriteria decision making* (MCDM)—includes methods for making trade-offs between multiple decision objectives, and for analyzing relationships between the ranking of decision alternatives and importance of different decision objectives. MCDM modeling can rely on traditional methods, which include multi-attribute utility theory and analytical hierarchy process. Aliah Think!, Criterion Decision Plus, Expert Choice Professional, and Logical Decisions for Windows (LDW) employ these traditional methods. Products like Decide Right and Which & Why use proprietary variations of them.

Model and data definition

Building decision models and specifying the necessary data is the most important aspect of using a DSS generator. Since the DSS generator applies its remaining functions (such as sensitivity analyses and the user interface mechanisms for using them) to the model schema, these functions become available without further cost once the model has been specified. Not having to provide these functions frees developers to concentrate on developing the application's model schemas.

Model schema definition. Again, the tools can be divided along the lines of the paradigm they employ: decision making under uncertainty and MCDM.

All modern products that employ decision making under uncertainty offer a visual modeling language for model development. This means that model building involves manipulating icons (for different types of model nodes), edges (for connecting nodes), and objects from drop-down menus. The capabilities that DSS generators provide are extremely useful: They make model development easy and also minimize typing errors.

In Analytica, DSS developers can create an influence diagram model by defining nodes, connecting them, and entering formulas in the appropriate boxes. To help in entering formulas, Analytica makes relevant variables and mathematical functions available in drop-down lists, as shown in Figure 1a.

Similarly, DATA lets DSS developers select built-in icons that represent decision nodes, chance nodes, and so on. PrecisionTree allows developers to build a decision tree graphically. It also provides a spreadsheet interface for viewing and manipulating data.

Products that support multicriteria decision making also provide a visual modeling language for model development. Initially, DSS developers visually set up the hierarchy of criteria. Then they can use several visual tools and representations to rate the alternatives against the criteria and enter the criteria weights.

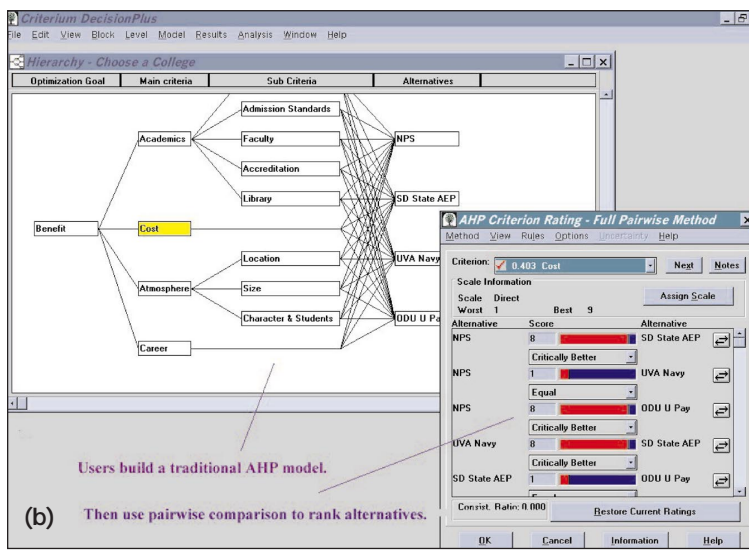
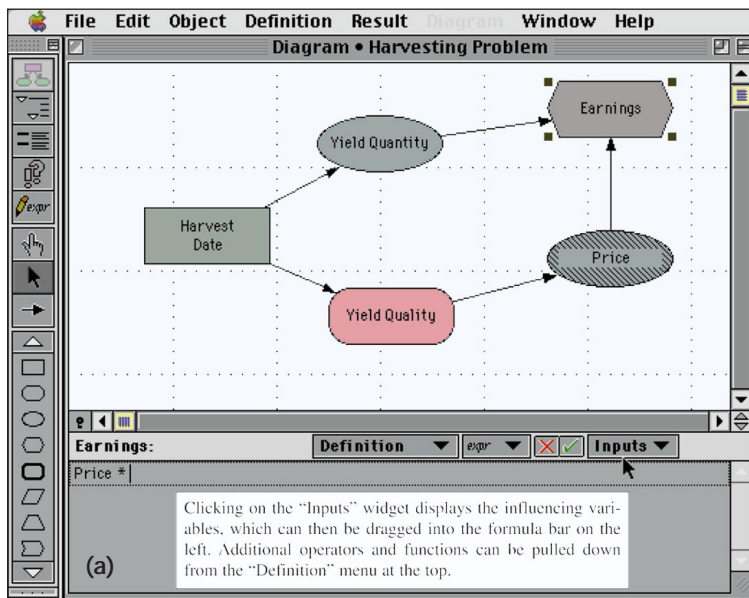


Figure 1. Model development and data input in (a) Analytica and (b) Criterion Decision Plus. In Analytica, DSS developers can drag suitable icons from the toolbar and manipulate them to build an influence diagram model. In Criterion Decision Plus, DSS developers can construct the hierarchy via graphical icons and then evaluate them via different mechanisms provided on menus.

The interface in Criterium Decision Plus, shown in Figure 1b, provides icons that DSS developers use to construct hierarchies. The generator automatically defines relevant mathematical variables and program elements such as data structures. The DSS generators automatically derive mathematical relationships and equations involving these criteria and alternatives.

All products help develop a fairly realistic and complex decision model within hours. They differ in the modeling paradigm they support and their modeling language. Here, a key differentiator is support for an algebraic modeling device called *indexing*.⁴ Indexing lets DSS developers group a collection of objects into a class (an index or an array), define variables and mathematical relationships that apply to all index elements, and declare aggregation functions (such as summation) over elements of an index variable.

Indexing structures⁴ are very useful in mathematical modeling, and their absence in most DSS generators is an important limitation. However, Analytica does explicitly support indexing in user-defined models.

Data definition. MCDM problems require decision-makers to rate the alternatives for each criterion. Thus a DSS generator requires a user to define this data in some fashion.

In some cases, these ratings are subjective and can be entered directly. Figure 2 shows an example from Expert Choice Professional. A DSS generator can also indirectly compute a rating via a user-specified pairwise comparison of alternatives (as in analytical hierarchy processing). In other cases, ratings may be derived as a function of available raw data values and a user-specified utility function.

Certain MCDM products (Decide Right, Which & Why, and Expert Choice Professional) accept only subjective user ratings. This restriction forces a user to implicitly determine a utility function, a limitation when measures are well defined and raw data is available. Here, LDW is unique in that it supports both methods as well as other multi-attribute functions, rating techniques, and elicitation tools (those that elicit user preferences).

To account for uncertainty in the decision problem, users can enter data values as single data points, tables, or as probabilistic distributions. (Commercial DSS generators support most parametric distributions.) The decision tree programs and a few of the MCDM programs accept data in this way. Analytica is distinctive in its use of multidimensional tables for data entry, display, and visualization.

From a user interface perspective, most products offer visual, direct-manipulation interfaces for entering model data such as probabilities, decision alternatives, utilities, and other values. Users typically access input forms for data by navigating the corresponding model schema element. Figure 1 and Figure

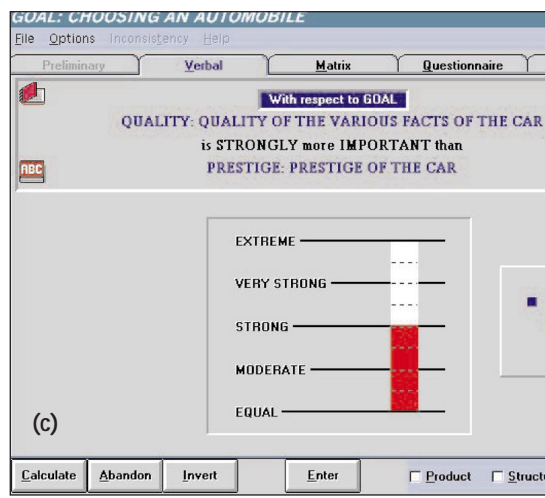
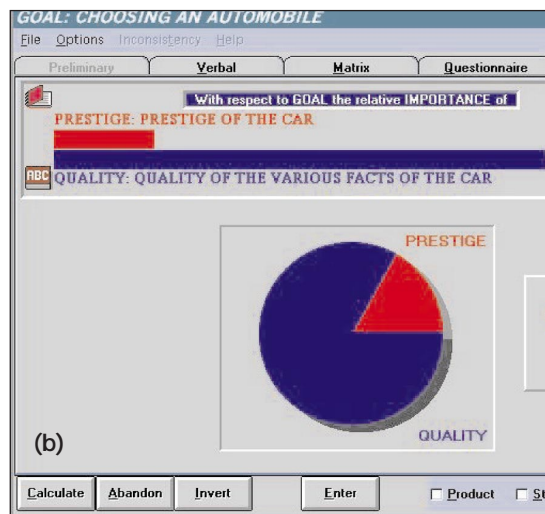
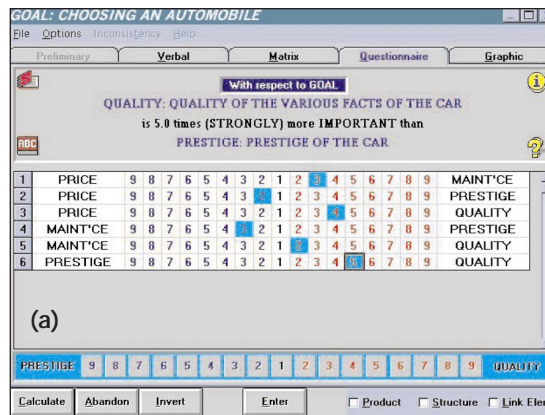


Figure 2. Data input in Expert Choice Professional. Once the decision model is set up, users can choose several methods for weighting criteria (or for grading alternatives): a) matrix method, b) graphical view, and c) textual method.

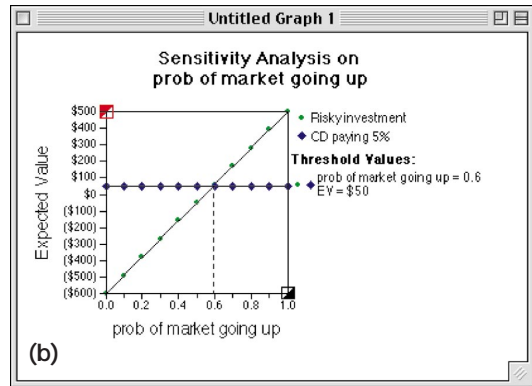
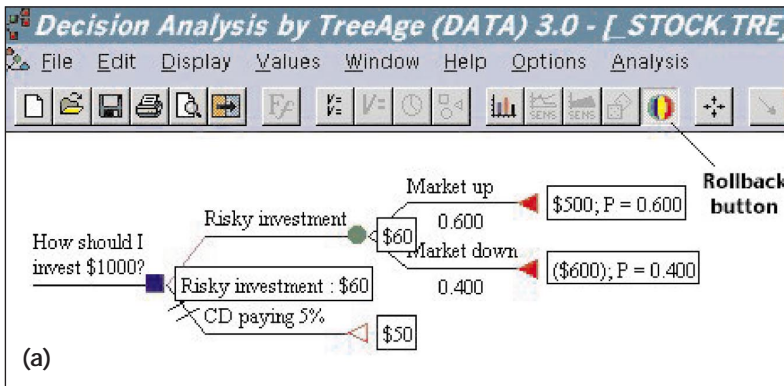


Figure 3. Decision tree analysis in DATA: Results of (a) tree rollback and (b) sensitivity analysis.

2 both show samples of these input forms. The drawback of this approach is that it fails to give the DSS developer a comprehensive view of the data. PrecisionTree, a spreadsheet add-in, is unique in this regard because it offers the familiar spreadsheet interface for data entry and manipulation.

Model and data management

Most commercial products work with one model of the decision problem at a time. Essentially, each model corresponds to a file or more than one file to store relationships, data, and so on. In this way, most products leave model management to the operating system, since only one model can be used at a time. Most don't offer additional functionality.

Analytica, however, is one exception. Users can build models by integrating existing models as modules in a "super" model. Several models can thus reuse the same modules.

Data management features within these products are fairly straightforward, and certainly less powerful than in database management systems. Data man-

agement in DSS generators doesn't enforce database design principles such as normalization, for example. In most cases, once data is entered, it is difficult to query or extract subsets of the data.

Analytical methods and visual representation

Effective use of models involves executing them repeatedly under different data scenarios and then analyzing the sensitivity of the results from an individual execution. The DSS generators discussed here offer a rich collection of commands and generic representations for analysis.

Basic output representations and dealing with uncertainty. In decision tree models, the basic output identifies the recommended decision branch and expected values (or other chosen measure) for each alternative. For example, in DATA, users simply click a "rollback" button to mark the superior decision branch on the tree (see Figure 3a). Users can deal with the uncertainty in input variables by using sensitivity analysis—Figure 3b displays an analysis of the probability of market behavior. Users can also have the application

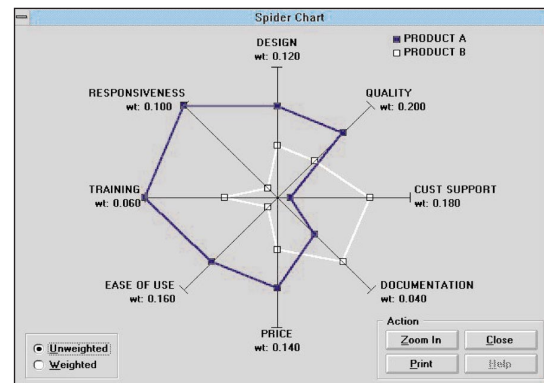
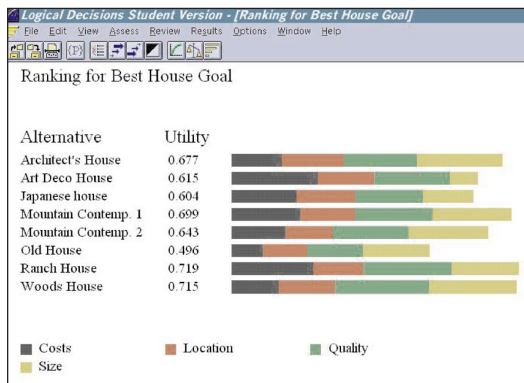


Figure 4. Different views of output in MCDM systems: In the (a) stacked bar chart (from LDW), each bar represents the score of an alternative, and stacks show the contribution from various criteria; (b) a spider chart (Alijah Think!) displays two or more alternatives based on various criteria.

distribute values probabilistically. Average-, worst-, and best-case results are available. Again, once a DSS developer specifies the model schema, these capabilities automatically become available to users of the specific DSS.

Programs for multicriteria problems typically display results as a bar chart showing the relative total scores for decision alternatives. A stacked bar chart shows, for each alternative, the contribution from each criterion. Scatter diagrams and spider diagrams provide additional formats for users in reviewing output. Figure 4 shows samples of two of these types of output.

Facilitating insight. An important purpose of decision support systems is to help users gain insight into the relationships among variables in the problem. Users must understand how changes in various controllable and/or uncontrollable variables influence the values of outcome variables.

Figure 5 displays two examples involving sensitivity analysis. In Figure 5a, LDW depicts the sensitivity of the final score to the weight of the price criterion. The graphic also displays the rankings of the alternatives and the cutoff points where these rankings change.

Figure 5b helps explain the way in which Expert Choice Professional provides dynamic sensitivity analysis: As users adjust the criteria weights (by moving the sliding teal bars at left), the software automatically updates the total scores of the alternatives (right). This gives users instant and graphical feedback on the relationship between final scores and criteria weights.

Finally in this category, we consider the output representation that Decide Right uses. As shown in Figure 6, Decide Right displays grades for each alternative for each criterion; color helps users quickly spot the comparative strengths and limitations of each alternative.

Not all people are comfortable with numbers and graphs or gleaning insight from several seemingly independent output representations. Textual outputs—*interpretations* of the numbers rather than mere translations into graphs or text—would be valuable to many users. Besides, in many situations, textual output is more portable than a collection of figures. Decide Right is distinctive in its attempt to translate and interpret quantitative analysis results into a textual representation.

User interfaces

The interfaces of the DSS generators we reviewed varied greatly, though most are relatively easy to learn and use. In most cases, however, and especially for complex problems, user interface design should account for at least *two* categories of DSS users:

- the DSS developer or analyst, who uses the generator to build a specific DSS; and

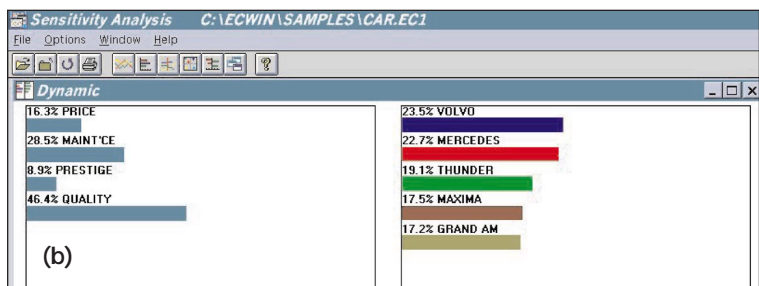
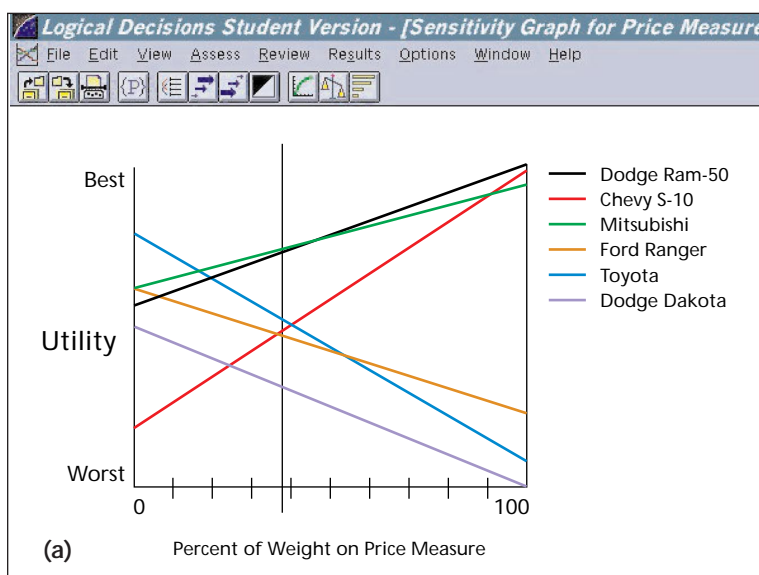


Figure 5. LDW provides (a) sensitivity analysis, and Expert Choice Professional provides (b) dynamic sensitivity analysis.

- the user or decision-maker, who must use the specific DSS created by the DSS developer.

You might expect user interface requirements—the representations, tools, commands, functions, and so on—to differ for these two categories of users. In addition, for many applications it can be desirable to build a *user flow* (a sequence of operations) into the user interface. This flow would guide the user at each step, rather than allowing only a *laissez-faire* approach to system use.

Unfortunately, most products do not distinguish between these two user roles and instead provide a general-purpose, but inflexible, user interface. DSS developers may find the user interface unsatisfactory for some applications. If they wish to extend it or to impose a certain user flow, they have no means of doing so, unless the product offers a good macro-programming language or can interface with external programs. Even then, imposing a user flow is usually quite difficult. As a result, commercial tools often choose to have users interact with the system via the same interface as the DSS developer. Not surprisingly, this interface can be too complex for users.

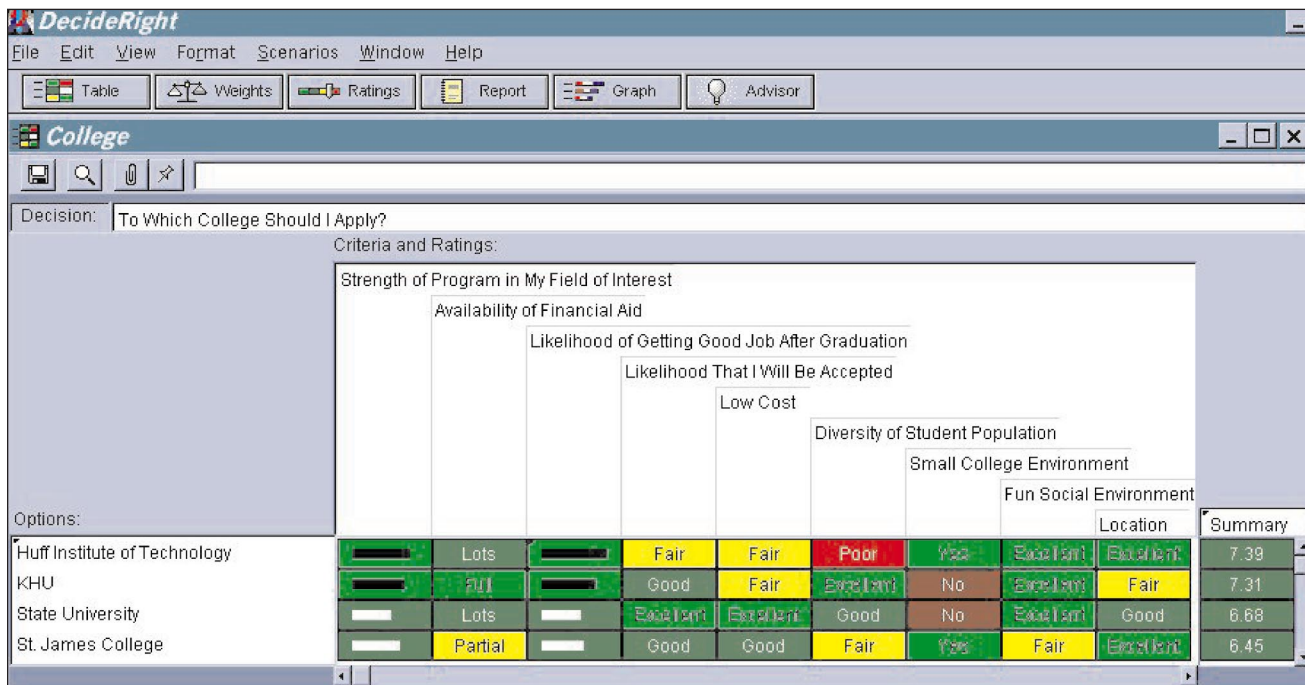


Figure 6. Decide Right provides this window to help users understand comparative strengths and weaknesses of alternatives. Color lets decision-makers see that the "State University" is a strong choice, weak only on the "small college" criterion.

A few products have modest tools for creating a separate user interface for users. DATA, for example, lets developers provide a custom interface. Such interfaces can contain a few buttons that users can use to enter data or to choose analysis types and graphs. Thus, users do not have to learn the complete (usually complex) interface provided for DSS developers.

Analytica offers a simple means of generating a very basic interface containing one-click buttons to obtain results and graphs. Also, products such as PrecisionTree operate within a spreadsheet, giving users an interface similar to the familiar spreadsheet user interface. In such a spreadsheet-coupled product, DSS developers also have access to all the usual spreadsheet macros and user interface design tools. These are significant advantages in creating custom interfaces for a specific application.

Model and data interchange

It is generally not possible to export model schemas from one product to another. This is not surprising since there are no standard formats for storing model schemas for different decision analysis model types. Some products do offer a form of version support. DATA and Decide Right, for instance, allow users to save different versions of a model and then recall them later. In most products, users must use the "save as" command to save different versions of the model as different file names.

On the other hand, most products can interchange data. Most DSS generators we reviewed could import and export data as well as create ASCII data files. Some also connected to popular desktop products such as spreadsheets and databases. These capabilities are useful, given the limited data design and management features in the DSS generators themselves.

FUTURE DIRECTIONS

Future decision support systems will probably support group decision making and will handle system components that are distributed over a client-server network. Only a few commercial products now support group decision making: Which & Why and Expert Choice Professional (via an extension called Team Expert Choice) let multiple users rank their preferences and combine these preferences into a group decision. These products also let groups assign weights to different decision-makers. Group features can be desirable, especially in higher level decision making, where decisions are frequently made by group consensus.

New client-server and distributed computing technologies on the Web can aid the remote access of DSS generators, providing decision support on demand.^{5,6} Here, the user's interface is a Web browser (the client), and separate components on a remote Web server provide data and model management features. Web-enabling a DSS allows a company to share complex and expensive DSS technologies at negligible extra cost, without customization for specific operating platforms. A few DSS generators already offer some form of Web-based interface and most vendors are paying more attention to this area.

Maturing DSS technologies and decision analysis methods have led to the development of DSS generators that are both usable and powerful. Yet the ultimate deployment of DSSs will depend on the ability of users to use them to their advantage. Clearly the needs of these users are important, so we are investigating this issue and have published preliminary results.^{7,8} Overall, we hope this article will spur the interest of IS managers, DSS vendors, researchers, and users in the use and development of DSSs and DSS generators. ♦

presents

Six Essential Short Courses in Web Technology, Software, and Technical Management

Winter/Spring 1999 on the UCLA Campus

Audience-Centered Web Design

March 1-3

With Kate Gomoll, Gomoll Research & Design

NEW COURSE

Web Form and Function: Visual Design for the Internet

March 4-5

With Kevin Mullet, Icarian

NEW COURSE

Developing Collaborative Applications on the Web

March 8-10

With Andreas Giergensohn, FX Palo Alto Laboratory;
and Alison Lee, IBM T.J. Watson Research Center

NEW COURSE

Legal Essentials for Protecting Software Development, Acquisition, and Use

May 6-7

With David G. Kay, University of California, Irvine;
and Michael M. Krieger, Bronson, Bronson &
McKinnon, LLP

NEW COURSE

Project Management for Development of Software Intensive Systems

May 10-12

With Donald S. Remer, Claremont Consulting Group

The 57th Technical Management Program

(formerly the Engineering and Management Program)

March 28-April 2

With 15 instructors

For more information about these and
other courses, call (310) 825-1047.
E-mail mhennes@unex.ucla.edu
Or access our Web site at:
www.unex.ucla.edu/shortcourses/

References

1. R. Sprague, "A Framework for the Development of Decision Support Systems," *MIS Quarterly*, June 1980, pp. 1-26.
2. A.L. Golub, *Decision Analysis: An Integrated Approach*, John Wiley & Sons, New York, 1997.
3. D. Buede, "Aiding Insight III," *OR/MS Today*, Aug. 1996, pp. 73-79.
4. A.M. Geoffrion, "Indexing in Modeling Languages for Mathematical Programming," *Management Science*, Mar. 1992, pp. 325-344.
5. H.K. Bhargava, R. Krishnan, and R. Muller, "Decision Support on Demand: Emerging Electronic Markets for Decision Technologies," *Decision Support Systems*, Vol. 19, 1997, pp. 193-214.
6. S. Sridhar, "Decision Support Using the Intranet," *Decision Support Systems*, Vol. 23, 1998, pp. 19-28.
7. C.L. Herrick, "A Survey of Software for Decision Analysis," master's thesis, Naval Postgraduate School, Monterey, Calif., Mar. 1997.
8. H.K. Bhargava, C.L. Herrick, and S. Sridhar, "Desirable Features for Decision Analysis Software," *Proc. Fourth Conf. Int'l Soc. for Decision Support Systems*, Int'l Soc. for Decision Support Systems, pp. 41-45.

Hemant K. Bhargava is a visiting professor at the Heinz School of Public Policy and Management, Carnegie Mellon University. He is on temporary assignment away from his position as an associate professor of information technology at the Naval Postgraduate School, Monterey, Calif. His research interests include decision support systems, e-commerce, and Internet-based electronic markets for decision technologies. Bhargava has a PhD in decision sciences from The Wharton School, University of Pennsylvania. He is a member of the IEEE Computer Society, the Institute for Operations Research and the Management Sciences (INFORMS), and INFORMS' Computing Society.

Suresh Sridhar is a consultant at i2 Technologies Inc. His research interests include decision support systems and distributed computing. Sridhar has a PhD degree in management information systems from Vanderbilt University. He is a member of the IEEE Computer Society and the ACM.

Craig Herrick is a lieutenant commander in the US Navy. His research interests include decision support and executive information systems. Herrick has an MS degree in information technology from the Naval Postgraduate School. He is a member of the US Navy Supply Corps.

Contact Bhargava at Carnegie Mellon University, The Heinz School of Public Policy and Management, 2109B, Hamburg Hall, Pittsburgh, PA 15213; bhargava@computer.org.