

Model-driven decision support systems: Concepts and research directions

Daniel J. Power^{a,*}, Ramesh Sharda^b

^a*University of Northern Iowa, Cedar Falls, IA 50614, USA*

^b*Oklahoma State University, Stillwater, OK 74078, USA*

Available online 20 July 2005

Abstract

In some decision situations, quantitative models embedded in a Decision Support System (DSS) can help managers make better decisions. Model-driven DSS use algebraic, decision analytic, financial, simulation, and optimization models to provide decision support. This category of DSS is continuing to evolve, but research can resolve a variety of behavioral and technical issues that impact system performance, acceptance and adoption. This article includes a brief survey of prior research. It focuses on model-driven DSS built using decision analysis, optimization, and simulation technologies; implementation using spreadsheet and web technologies; issues associated with the user interface; and behavioral and technical research questions. © 2005 Elsevier B.V. All rights reserved.

Keywords: Quantitative models; Decision support systems; DSS concepts; Research questions

1. Introduction

Given the growing complexity and uncertainty in many decision situations, helping managers use quantitative models to support their decision-making and planning is an important research topic. For more than 50 years economists, psychologists, operations researchers and management scientists have investigated this topic from their various perspectives, but researchers have only just begun to understand the behavioral and

technical challenges of designing, developing and implementing effective model-driven Decision Support Systems (DSS).

By definition one or more quantitative models are the dominant components that provide the primary functionality of a model-driven decision support system [65]. Also, by definition a model-driven DSS is designed so a user can manipulate model parameters to examine the sensitivity of outputs or to conduct a more ad hoc “what if?” analysis. Two characteristics differentiate a model-driven DSS from the computer support used for a decision analytic or operations research special decision study: (1) a model in a model-driven DSS is made accessible to a non-technical specialist such as a manager through an easy to

* Corresponding author.

E-mail addresses: power@dssresources.com (D.J. Power), sharda@okstate.edu (R. Sharda).

use interface, and (2) a specific DSS is intended for some repeated use in the same or a similar decision situation. The general types of quantitative models used in model-driven DSS include algebraic and differential equation models, various decision analysis tools including analytical hierarchy process, decision matrix and decision tree, multi-attribute and multi-criteria models, forecasting models, network and optimization models, Monte Carlo and discrete event simulation models, and quantitative behavioral models for multi-agent simulations. Models in a model-driven DSS should provide a simplified representation of a situation that is understandable to a decision maker [9,31,64,77].

Model-driven DSS are continuing to evolve, but additional research needs to be conducted. The objective of this review is to highlight recent research related to model-driven DSS and identify research needs and directions. The next section briefly summarizes an expanded framework and basic concepts associated with computerized decision support systems. Section 3 focuses on an overview of model-driven DSS issues and prior research. Also, Section 3 discusses research about model-driven DSS applications, underlying modeling techniques, delivery mechanisms, and the DSS user interface. Section 4 identifies research directions related to behavioral and technical aspects of developing, implementing and using model-driven DSS. The final section summarizes and concludes the analysis and review.

2. DSS framework and constructs

Categorizing decision support systems can assist researchers and managers in understanding how this general class of information systems impacts decision behavior and how one should design and construct such systems. The expanded DSS framework developed by Power [62,64,66] provides a means of differentiating decision support systems. The framework extends the terminology, definitions and concepts from prior frameworks and theory. This analysis focuses on only the model-driven DSS category from the expanded DSS framework. In the DSS literature, this category of applications has been variously identified as model-oriented, model-based, or quantitative DSS, when and if an attempt was made to differentiate

and categorize a specific DSS that was included in a research study.

The expanded DSS framework specifies five categories of DSS that can be recognized by identifying the dominant architectural component that provides the functionality for supporting decision-making [64]. The five categories include model-driven DSS, as well as communications-driven, data-driven, document-driven and knowledge-driven DSS. The framework specifies three secondary dimensions that are also relevant to model-driven DSS: the purpose of the DSS, the intended users of the DSS, and the enabling technology used to implement the DSS artifact. The expanded framework builds upon Alter's [3] seven types of DSS and Sprague's [76] components of a DSS, including model, database, architecture and user interface.

Model-driven DSS include computerized systems that use accounting and financial models, representational models, and/or optimization models to assist in decision-making. Model-driven DSS emphasize access to and manipulation of a quantitative model and hence the model or models are the dominant component in the DSS architecture that provides the functionality for the DSS. Simple analytical tools based upon algebraic models provide an elementary level of functionality. Model-driven DSS use data and parameters provided by decision-makers to help in analyzing a situation, but such systems are not data intensive.

Communications-driven DSS derive their functionality from communications and information technologies that are used in the system to support shared decision-making. Data-driven DSS include file drawer and management reporting systems, data warehousing and analysis systems, Executive Information Systems (EIS) and data-driven Spatial DSS. Business intelligence systems are also examples of data-driven DSS. The functionality of this category of DSS results from access to and manipulation of a large database of structured data. Document-driven DSS integrate a variety of storage and processing technologies to provide sophisticated document retrieval and analysis to support decision-makers. Finally, knowledge-driven DSS suggest or recommend actions based upon knowledge that has been stored using Artificial Intelligence or statistical tools like case-based reasoning, rules, frames and Bayesian networks. The knowledge

component provides the primary functionality for the decision support system or subsystem.

An information system may include both transaction processing and decision support subsystems. Also, integrated systems may have been developed to primarily provide a specific type of decision support, but other decision support subsystems may also be included. For example, an information system may include a model-driven DSS and it may also include a knowledge-driven DSS module for pre- or post-processing. Similarly, a DSS may include both a data-driven and a model-driven subsystem. In this paper, our focus is on DSS that primarily derive functionality from one or more quantitative models, though the application may have other decision support subsystems.

Decision support researchers also need to differentiate three computerized systems associated with improving or enhancing individual and organizational decision-making (see Fig. 1). Automated decision systems are intended to automate and make decisions in routine, well-structured situations, whereas decision support systems are auxiliary or ancillary systems intended to assist decision-makers in a wide variety of semi-structured and recurring decision situations. Finally, computerized tools used by technical experts to complete special decision studies are usually not appropriately categorized as decision support systems. These systems have user interfaces intended for expert users and the computerized system is intended for use only for the specific specialized study. In general, computer support developed for a specific specialized study by staff experts or computer support staff is not built to support a specific decision process or built for more ad hoc analysis that might occur on a recurring basis. For example, Clemen and Kwit [15] acknowl-

edged the “one-time nature of typical decision analysis projects”. Thus the focus in this analysis is on systems for repeated use that include accessible user interfaces for non-technical people to support their decision-making.

3. An overview of model-driven DSS research

Given that there is still some disagreement about the types of DSS and that there is a vast amount of research related to model-driven DSS, this section attempts to offer only a representative sample of prior model-driven DSS research. The intent is not to provide an exhaustive list but rather to provide an overview and analysis of model-driven DSS research. Model-driven DSS applications have been reported for all business functional areas, general management tasks, and for tasks as diverse as cow culling and natural resource management. Eom [22,23] identified a wide variety of DSS applications reported in the academic literature. For the period 1970–1992, Eom identified 474 DSS applications reported in the literature, excluding conference proceedings papers and doctoral dissertations. Also, Eom [23] identified more than 1800 DSS related articles and many of them focused upon what this article defines as model-driven DSS. Kim and Eom [40] provide an additional bibliography of DSS applications. Most issues of *Decision Support Systems* and *Interfaces* include articles describing realistic applications of DSS. Recently, Shim et al. [72] provided a broad conceptual perspective on future directions for decision support systems development. Our goal is to highlight examples of research streams and applications, with a particular focus on recent industry applications and technology innovations.

Algebraic models are perhaps used most frequently in building model-driven DSS applications, especially those developed within spreadsheets. In general, personal computer-based spreadsheet technology has been robust and reasonably understandable for building model-driven DSS. Research about specific DSS built using financial models in a DSS generator like Interactive Financial Planning System (IFPS) was common for many years [71]. In recent years however, that research stream has become less active because spreadsheet software replaced systems of the

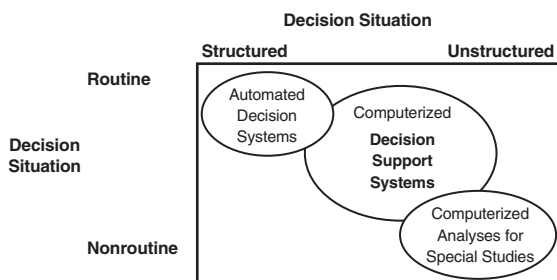


Fig. 1. Differentiating computerized support for various decision situations.

IFPS genre. Today, three more complex techniques are being used to build model-driven DSS: decision analysis, mathematical programming, and simulation. The next three sub-sections focus on recent applications of these quantitative techniques in building model-driven DSS. Then DSS development delivery mechanisms and user interface issues are briefly assessed.

3.1. Decision analysis

Decision Analysis (DA) broadly refers to methods that involve quantified evaluations of possible alternative courses of action. The evaluations often include an assessment of probabilities and preference elicitation using direct or indirect utility functions. There is some debate about whether specific techniques belong in the decision analysis domain. For example, Keefer et al. [39], in reviewing DA applications of the past decade, exclude applications involving the analytic hierarchy process and multi-criteria techniques. They cite Mollaghasemi and Pet-Edwards [55] and Yoon and Hwang [86] for details on these methods. This analysis uses the broadest, most inclusive, definition of decision analysis techniques. Keefer et al. [39] review applications of DA methodologies, specifically decision trees and influence diagrams in diverse industries such as energy, manufacturing, finance, project selection, medical, military, and public policy decisions.

Podinovski [61] describes a DSS for applying multiple criteria decision analysis (MCDA) for eliciting a decision maker's value structure. He reports an application of the DSS in an environmental management problem. Ülengin et al. [81] developed a hierarchical, model-driven DSS to help make decisions on reducing congestion in the Bosphorus Sea. The objectives of 19 experts from different backgrounds were captured to build a hierarchical cognitive map, and then analyzed in a decision matrix. Dunning et al. [21] report a special study applying strategy tables, influence diagrams and decision trees to develop a multi-year schedule for refueling a nuclear power plant for the New York Power Authority. Aldag and Power [2] studied a general process-oriented decision analysis decision aid. Power et al. [67] studied the impact of using a tool kit of decision analysis techniques on individual decision behavior.

Many decision analysis software tools and specific DSS applications built using the tools have been reported in the academic as well as the practitioner literature. For example, Proctor and Gamble used Precision Tree (www.palisade.com) to make decisions on site locations. Precision Tree is a spreadsheet add-in to permit integration of influence diagrams and decision trees. Expert Choice (www.expertchoice.com), an implementation of the Analytic Hierarchy Process (AHP), has been used in many DSS research studies. Catalyze (www.catalyze.co.uk) markets two packages, Equity and Hiview that have been used to operationalize decision trees and preference elicitation methods. Specific model-driven DSS built using these DSS generators have been studied. Other decision support generator tools based upon decision analysis techniques include Logical Decisions (multiple objectives), Super Tree, and Tree Plan. Analytical Decision Engine (www.lumina.com) provides a tool kit to customize DSS applications incorporating DA capabilities with domain specific user interfaces. *OR/MS Today* publishes surveys of such software regularly [52]. This research area continues to grow both in terms of available software generators to build DSS as well as in the reporting of interesting applications in new domains.

3.2. Optimization and mathematical programming models

Many model-driven DSS are intended to provide some type of optimization of desired decision criteria. Shim et al. [72] describes the state of the art of DSS including optimization-based DSS. *OR/MS Today* publishes reviews of optimization (especially linear programming) software [28] periodically. Such software is used in building DSS to support specific applications. Examples of such DSS generators include the AIMMS Modeling System (www.paragon.com), ILOG Optimization suite (www.ilog.com), MPL (www.maximalsoftware.com), etc. Again, the reader is referred to Fourer [28] for a comprehensive listing of such tools.

Although model-driven DSS based upon optimization have been deployed in many settings, two application categories have become important recently: revenue management and supply chain management. Revenue management or yield management is aimed

at identifying optimal price levels of services by analyzing forecasts and current sales levels [59]. Smith et al. [74] provide an overview of initial modeling efforts at American Airlines. Marsan [50] provides an overview of such systems in the hospitality industry. Boyd and Bilegan [11] provide a recent overview of revenue management research and practice in many industries. DSS built for revenue management have become a competitive necessity [58]. Butchers et al. [14] describe eight optimization-based DSS intended to help optimize crew scheduling including “tours-of-duty planning and rostering.”

Supply chain management has become a major area for decision support applications with the growth of Enterprise Resource Planning (ERP) applications in organizations; because of ERP it has become easier to get data needed to model supply chains. Decision support models are now available for various stages of supply chain management, including logistics planning, production planning, demand management, and pricing decisions [48,49]. For example, Smith et al. [75] describe a DSS for supply chain planning to allow retailers who sell private-label products to manage their sourcing allocations. Vendors who supply this private-label merchandise to the retailer may have different lead-time requirements, pricing, and production capabilities. Smith et al. developed a model-driven DSS to develop sourcing plans that maximize the retailer’s expected gross profit. Vigus et al. [83] studied the optimization and financial model-based planning systems at the Kellogg Company. Their study reported multi-million dollar savings from using the computerized planning and operational systems.

An emerging supply chain application for model-driven DSS is termed demand optimization or demand chain management [46]. The idea is to employ optimization models that incorporate uncertainty, product relationships, and stock levels to decide prices for thousands of products that a retailer may have. Some software vendors have developed model-driven DSS that may include stochastic programming, integer programming, and modeling language interfaces to enable rapid model modification, and large-scale data integration capabilities. McClain [53] reports that such consumer demand management DSS are beginning to be used in companies such as Radio Shack, Cargill, and Duane Reade (a drug chain). Model-

driven DSS have enabled companies to price their products “strategically.” Such applications are likely to expand as companies are better able to take advantage of large databases, high bandwidth networking to propagate the data in real time, and faster computers to enable solutions of very large models.

Kim and Eom [40] list many recent applications of optimization-based DSS. For example, Katok and Ott [38] describe a DSS based on mixed-integer programming to determine weekly production schedules for Coors aluminum cans. This DSS is implemented in a spreadsheet environment. Another spreadsheet-delivered optimization DSS is reported by LeBlanc et al. [45] in assigning managers to construction projects. Other examples include DSS for timetabling decisions [27].

Constraint logic programming is emerging as another major technique for optimization DSS. ILOG (www.ilog.com) and others provide software and tools to build such DSS. One such DSS generator is described in Fierbinteanu [26].

3.3. Simulation techniques

Simulation is a broad term that refers to an approach for imitating the behavior of an actual or anticipated human or physical system. The terms simulation and model, especially quantitative and behavioral models, are closely linked. A model used in a simulation can capture much detail about a specific system, but how complex the model is or should be depends upon the purpose of the simulation that will be “run” using the model. Both simulation special studies and model-driven DSS built using simulation techniques involve multiple experiments or “runs” of the simulation. The results of each “run” are recorded and then the aggregate results are analyzed and manipulated to try to answer specific decision related questions.

There are several types of simulation and a variety of overlapping terms are used to identify them: Monte Carlo simulation, traditional mathematical simulation, activity-scanning simulation, discrete simulation, event-driven simulation, probabilistic simulation, process-based simulation, real-time simulation, data-driven simulation, agent-based and multi-agent simulation, time dependent simulation, and visual simulation [24,43,56].

In a Monte Carlo or probabilistic simulation one or more of the independent variables is specified as a probability distribution of values. A probabilistic simulation helps take risk and uncertainty in a decision situation into account in the results. Time dependent or discrete simulation refers to a situation where it is important to know exactly when an event occurs. For example, in waiting line or queuing problems, it is important to know the precise time of arrival to determine if a customer will have to wait or not. According to Evans and Olson [24] and others, activity-scanning simulation models involve describing activities that occur during a fixed interval of time and then simulating for multiple future periods the consequences of the activities while process-driven simulation focuses on modeling a logical sequence of events rather than activities.

Simulation can assist in either a static or a dynamic analysis of a system. A dynamic analysis is enhanced with software that shows the time-sequenced operation of the system that is being predicted or analyzed. Simulation is a descriptive tool that can be used for both prediction and exploration of the behavior of a specific system. A complex simulation embedded in a model-driven DSS can help a decision maker plan activities, anticipate the effects of specific resource allocations and assess the consequences of actions and events [4].

In some decision situations, simulation specialists build a simulation and then conduct a special empirical study and report their results to management. Evans and Olson [24] discuss examples of how simulation has been used to support business and engineering decision-making. They report a number of special decision support studies including one that evaluated the number of hotel reservations to accept to effectively utilize capacity to create an overbooking policy, a Call Center staffing capacity analysis, a study comparing new incinerating system options for a municipal garbage recycling center, a study evaluating government policy options, and various studies for designing facilities. Examples of model-driven DSS built with a simulation as the dominant component include: a Monte Carlo simulation to manage foreign-exchange risks; a spreadsheet-based DSS for assessing the risk of commercial loans [16], a DSS for developing a weekly production schedule for hundreds of products at multiple plants; a program

for estimating returns for fixed-income securities; and a simulation program for setting bids for competitive lease sales [24].

Agent-based or multi-agent simulations are supplementing traditional simulation techniques. In the past 5 years, agent-based visual simulations have become an alternative approach for analyzing some business systems. According to Bonabeau, founder of Icosystem Corp., in Rothfeder [70], “People have been thinking in terms of agent-based modeling for many years but just didn’t have the computing power to actually make it useful until recently. With agent-based modeling, you describe a system from the bottom up, from the point of view of its constituent units, as opposed to a top-down description, where you look at properties at the aggregate level without worrying about the system’s constituent elements.”

Multi-agent simulations can be used to simulate some natural and human-created systems where traditional simulation techniques may be ineffective. Examples of systems that can be simulated and analyzed with a model-driven DSS based upon a multi-agent simulation include a grocery store with shopper and worker agents, a wholesale market with buyers and sellers, an airport with passengers, visitors and employees, and a factory with production workers and supervisors.

A behavioral model of an agent is an explicit quantitative statement of variables that impact the observed actions of a system of objects or of a specific object or entity [12]. A behavioral model is used to help understand, explain, and predict behavior. Behavioral models are usually specified as mathematical equations or as computer programs, rather than as verbal descriptions. A behavioral model is built by observing the previous behavior of an entity or a system; the resulting model can then be used to predict future behavior and performance. Microsimulations of people have also been used to make healthcare decisions. For example, van der Ploeg et al. [82] describe STDSIM, a microsimulation model to assess the effects of different approaches to manage sexually transmitted diseases (STD). The reported application was in Kenya, but the model can be generalized for use elsewhere.

In realistic, visual simulations [4] many models may be needed to “drive” the simulation. Models of the physical environment ensure that natural laws are not violated. For example, the simulation “logic” may

specify that two objects cannot occupy the same space. From a decision support perspective, the really interesting simulations are those that help a decision maker anticipate human behavior of customers, voters, or enemy soldiers. These simulations need to imitate physical reality, but more importantly multiple “human-like” agents need to be included in the simulation. Currently this is happening in two ways. Some simulations use “real actors” who make choices in a simulated on-line environment. A multi-player simulation like “America’s Army” (www.americasarmy.com) is an example of this approach. Another approach is to use behavioral models as the “actors/agents” that are making choices in the simulated environment. We can refer to these approaches as multi-player and multi-agent simulations. Today a multi-player simulation usually also includes some computer-based agents.

Gimblett et al. [29] reported a project to develop a decision support and simulation system that used autonomous agents to assist natural resource managers in assessing and managing dynamic recreation behavior, social interactions and resulting conflicts in wilderness settings. They linked dMARS (Distributed Multi-agent Reasoning System) [20], the Swarm Multi-agent Simulation System and a GIS system to develop the model-driven DSS. They calibrated the autonomous agents using survey data from people using a recreation facility in Sedona, Arizona. The realistic simulation was intended to support forest management activities and assist in evaluating proposed practices for recreation use in the recreation facility. Another example of combining GIS and simulation in a DSS is given by de Silva and Eglese [19] for evacuation planning. Lee et al. [47] studied a pricing DSS that integrated simulation of a wholesale fish market with a regression model to help decision-makers set daily wholesale prices.

Some of the major providers of simulation-based DSS generators include Goldsim, Imagine That, Inc., Decisioneering, and Palisade software. *OR/MS Today* publishes reviews of various simulation tools. For the most recent survey, please see Swain [79].

3.4. DSS development and delivery mechanisms

This section reviews the development and delivery processes for building model-driven DSS. Many early

DSS were built using general-purpose development software. Today most model-driven DSS are built using the quantitative techniques described in the previous section using DSS generators [77]. Early versions of these software systems allowed a more user-friendly development of a DSS, but the systems were still run and maintained on a central system. One such system that was mentioned earlier is IFPS, used in many commercial applications and academic studies [71].

Spreadsheets clearly have become the most ubiquitous DSS generators. Even specialized quantitative techniques such as optimization are usually implemented using a spreadsheet. Section 3.4.1 reviews developments in building spreadsheet-based DSS. Typically, model-driven DSS that are intended for only a few users are made available as stand-alone applications. DSS that may be widely used or may be used by multiple units of an organization are beginning to be offered as web-based DSS. Also, some model-driven DSS are deployed for use by multiple, interacting teams of users. Historically, these systems have been called Group DSS (GDSS). Sections 3.4.2 and 3.4.3 review these developments in deploying model-driven DSS.

3.4.1. Spreadsheet-based DSS

A Decision Support System that has been or will be implemented using a spreadsheet package can be termed a spreadsheet-based DSS [69]. A spreadsheet is one of the major enabling technologies for deploying model-driven DSS. A wide variety of decision support purposes can be achieved using a spreadsheet package, e.g., Microsoft Excel.

In the world of accounting, a spreadsheet “spreads” or shows data on a single sheet of paper for a manager to look at when making a decision. Also, a spreadsheet is a collection of cells whose values can be displayed on a computer screen [63]. By changing cell values and having all cell values re-evaluated, a user performs “what if?” analysis and can observe the effects of those changes. Spreadsheet packages are DSS generators. Sprague and Carlson [77] defined a DSS generator as “a computer software package that provides tools and capabilities that help a developer quickly and easily build a specific Decision Support System (see p. 11).” Spreadsheet packages qualify as DSS generators because: (a) they have sophisticated

data handling and graphic capabilities; (b) they can be used for “what if” analysis; and (c) spreadsheet software can facilitate the building of a DSS.

Model-driven and data-driven DSS are the primary types of DSS one might consider developing using a spreadsheet package. Spreadsheets seem especially appropriate for building a DSS with one or more small models. A developer implements the model and then adds buttons, spinners and other tools and representations to support a decision maker in “what if?” and sensitivity analysis. A data-driven DSS can also be implemented using a spreadsheet. A large data set would be downloaded to the DSS application from a DBMS, a web site or a delimited flat file. Then pivot tables and charts could be developed to help a decision maker summarize, manipulate and understand the decision relevant data.

Spreadsheet-based DSS can be created in an end user development environment or in a multi-user environment. Microsoft Excel is probably the most popular spreadsheet application development environment today, though DSS have been built using other spreadsheet software such as Lotus 1–2–3. Spreadsheet-based DSS have become so common that most modeling courses now are based on using spreadsheets [68]. Some examples of spreadsheet-based DSS tools are discussed in the next paragraph.

Frontline Systems (<http://www.solver.com> and <http://frontsys.com>) provides the optimization add-in or “Solver” that is packaged with Excel, but they also sell Premium Solver. It is a more powerful program and comes with example spreadsheet models and a user guide. The major product of Decisioneering, Inc. is Crystal Ball. It is a suite of Excel-based risk analysis, optimization and forecasting tools that can be used to build specific applications (<http://www.crystalball.com>). Palisade Software (<http://www.palisade.com>) markets the @RISK add-in. The Palisade DecisionTools Suite includes five products: @RISK, PrecisionTree®, TopRank®, BestFit®, and RISKview™. Other optimization add-ins include GeneHunter (<http://www.wardsystems.com>), Evolver (<http://www.palisade.com>), What’s Best (<http://www.lindo.com>) and XPRESS (<http://www.dash.co.uk>). Decision Support Services has a product called Decision ToolPak (<http://www.decisiontoolpak.com>). It consists of three decision modeling add-ins for Microsoft Excel: TreePlan, SensIt and RiskSim. XLSIM (<http://www.analycorp.com>)

is a Monte Carlo Simulation add-in for Excel. Simtools.xla (<http://home.uchicago.edu/~rmyerson/addins.htm>) adds statistical functions and procedures for doing Monte Carlo simulation and risk analysis in spreadsheets.

3.4.2. Web-based DSS

When vendors propose a Web-based DSS they are referring to a computerized system that delivers decision support information or decision support tools to a manager or business analyst using a ‘thin client’ Web browser. Vendors and MIS practitioners are making some distinctions about the technology platform used to deliver decision support that should be noted. Most notably the phrase “Web-enabled” has crept into the DSS lexicon. It is important to understand how a Web-based DSS differs from a Web-enabled DSS.

When the enabling technology used to build a DSS is the Internet and Web, it seems appropriate to call the system a Web-based DSS. Web-based should mean the entire application is implemented using Web technologies including a Web server, HTML, CGI, PHP, and possibly a database product like Oracle 10g or MySQL; Web-enabled means key parts of an application such as a database remain on a legacy system, but the application can be accessed from a Web technology component and displayed in a browser.

Some legacy DSS can be Web-enabled much faster and at a much lower cost than would be possible if the DSS were redeveloped using Web technologies. But, many of the benefits of a Web-based DSS can be realized by implementing a Web-enabled DSS. So a Web-enabled DSS may be the best choice for making an existing DSS more widely available.

Web technologies can be used to implement any category of DSS including communications-driven, data-driven, document-driven, knowledge-driven, and model-driven DSS. At one point, most systems labeled “Web-based DSS” were linked to a data warehouse, but that is certainly no longer the case. A model-driven decision support simulation developed in Java can be delivered via the Web and so can a large HTML/XML text repository that is part of a document-driven DSS. With a Web-based or a Web-enabled DSS no particular decision support software needs to be on the client computer. A Web browser and an Internet connection deliver the decision support functionality to the user. Kuljis and Paul [43] and

Miller et al. [54] reviewed web-based simulation research and developments.

A new generation of Web-based model-driven DSS is beginning to emerge. These systems take advantage of the web services concept. A single DSS problem may be solved using multiple modeling or solution paradigms developed by different sources as separate web services and the results are then presented to the user in an aggregated or summarized form. An example of this new generation of DSS is described in Delen and Sharda [17]. Their model-driven DSS called Movie Forecast Guru allows a decision maker to generate a forecast for a movie's box office performance by changing the movie's decision parameters (star power, season of release, special effects, etc.). Changes in these parameters are analyzed by a number of trained forecasting models based upon techniques such as neural networks, logistic regression, and discriminate analysis. Each model works as an independent expert called upon to provide advice; implemented as a web service. The final results are presented to the decision maker from each expert as well as in an aggregated form. The system was implemented using the .NET framework. More innovative web-based, model-driven DSS of this type are likely to be developed in the future.

3.4.3. Group model-driven DSS

Group Decision Support Systems (GDSS) have been studied extensively and that research stream remains very active in the literature. DeSanctis et al. [18] provide an overview of research issues in GDSS. The primary focus of this research is about improving the process of group decision-making. However, some GDSS deliver model-driven DSS to a group of users. Nunamaker et al. [57] focus on model management and group decision support. Companies such as Group Systems (www.groupsystems.com) offer technologies and process support to use models to support group decision-making. This approach has also been termed decision conferencing. Bresnick et al. [13] describe a military application of decision conferencing. Catalyze (www.catalyze.co.uk) also offers decision conferencing services in conjunction with multi-criteria decision analysis. A combination of these process delivery services and the use of sophisticated model-driven DSS should encourage even greater use of model-driven GDSS. Team Expert Choice from

Expert Choice, Inc. (www.expertchoice.com) is an example of a generator for building specific group model-driven DSS based upon the Analytical Hierarchy Process (AHP). The case study by Wasyluk and Saaty [85] illustrates the use of a model-driven GDSS in the U.S. Department of Veteran's Affairs.

3.5. DSS user interfaces

The goal of making model-driven DSS accessible to non-technical specialists implies that the design and capabilities of the user interface are important to the success of the system. A model-driven DSS user interface provides capabilities for inputting values, for manipulating values and of equal importance, the user interface controls how the user views results and influences how the user understands results and hence influences choices. DSS designers and researchers need to assess how the anticipated users of the DSS should enter values. Also, it is important to evaluate the order of input fields and stimuli given to the user about what values are sought. The DSS interface design can also bias users and inappropriate elicitation approaches are sometimes used.

In building a model-driven DSS, a designer needs to be concerned about eliciting certain and uncertain quantities and qualities, objective and subjective probabilities, utilities, and weights. It may be necessary to elicit probability point estimates, probability distributions, utility functions, monetary values and monetary estimates, preferences, integer quantities, distances, scale values, and priorities. Values can describe objective and subjective measures of concrete objects and appraisals of feelings, beliefs and attitudes. Values may be estimated or based on actual measurement using a scale. The scales may involve physical or perceptual units.

Values are elicited from a decision maker, assessor, estimator or appraiser—the user of a model-driven DSS. In general, a question or another type of stimulus indicates what value is being elicited. Values are elicited as part of a valuation or elicitation process. The elicitation approach in a specific DSS may reduce or increase errors in the values that are obtained. The three primary approaches for eliciting values are: (1) numerical, (2) graphical, and (3) verbal elicitation.

Guerlain et al. [30] examined what types of information should be displayed with what representations.

Also, they examined when visual representations are better than text-based displays. According to Guerlain et al., “Each representation makes some information about a problem salient while making other information more difficult to see (p. 26)”. More research is needed on these issues for a variety of types of DSS, the Guerlain et al. results may only hold for a data-driven DSS interface for technical operators.

Apparently some people can process certain types of value information much more effectively using visual displays and graphical input modes than they can process them numerically or verbally. Each approach for eliciting values has strengths and weaknesses that must be better understood and that a DSS software designer must recognize. Current evidence suggests that unbiased values can be elicited [44,84]. The appropriate user interface design seems to depend on the task and the skill and training of the user.

4. Model-driven DSS research needs

Behavioral and technical research on model-driven DSS needs to address many unresolved issues associated with construction of specific quantitative models, storage and retrieval of data needed by different types of models, communication of parameters among models and other DSS components, multi-participant interaction in model use and value elicitation, and the impact of user interface design alternatives on model-driven DSS effectiveness and ease of use. Also, researchers need to investigate issues associated with building, deploying and using model-driven DSS. This broad listing of needs seems daunting, but it suggests more specific research issues and questions for further research. The spectrum of possible model-driven DSS is already broad and as we learn more the implementation of model-driven DSS will expand to incorporate new decision situations and new modeling approaches. The remainder of this discussion of model-driven DSS research needs is divided into fourteen behavioral and technical questions that most seem to warrant further study.

4.1. Behavioral questions

The following list of research questions focuses on topics related to understanding the behavioral impact

of model-driven DSS. The questions seem largely unresolved in the current literature, but we have in some cases identified relevant prior research.

B1. Are users of a model-driven DSS who understand the model more likely to appropriately use the results? When they understand the model, are they more frequent users? Also, do they have more confidence in the results? When users understand the model, are the systems more effective in improving decision-making?

The presumption has been that managers need to understand any quantitative models in a model-driven DSS to benefit from using the system. Researchers in many empirical studies have noted that inadequate research has investigated the above questions [2,36,41,71]. No one study will demonstrate that all model-driven DSS improve decision-making. Rather it is important to conduct meta studies that examine contingency factors including a user’s understanding of the underlying models in prior model-driven DSS research.

B2. Do some users of a model-driven DSS attempt to bias or manipulate model parameters using “What if?” analysis to yield specific results? If so, what types of users are more likely to misuse the “What if?” capability? Also, what conditions impact the biased use of a model-driven DSS?

The general perception is that some decision makers bias their use of a model-driven DSS to obtain the “answer” that is desired. In particular, because a computerized DSS facilitates “What if?” analysis, some observers have concluded that biasing is easy and facilitated by a model-driven DSS. This perception is supported by some empirical research linked to behavioral decision theory [36,84]. Also, it is important to observe under what conditions biasing occurs when it does occur and whether features of a specific DSS encourage or discourage biasing. The phenomenon has not been adequately explored in the context of model-driven DSS.

B3. Can some design alternatives and value elicitation methods in a model-driven DSS user interface reduce the occurrence of biased decision behavior?

Research associated with elicitation of subjective probabilities and values suggests that de-biasing can

occur and that some elicitation techniques produce normatively better results than do others [1]. Most of the research associated with elicitation of values including elicitation of subjective probabilities has occurred using non-computerized interventions [6,84]. Because the elicitation of values can impact the effectiveness of a model-driven DSS more research on elicitation of values needs to be conducted. The studies need to examine a wide variety of elicitation approaches, including numeric, graphical and verbal, in the context of various models.

B4. Does a model-driven DSS user interface customized to individual user differences impact the subsequent use of a DSS? How much “customization” is needed and possible?

Personalization of Web portals and other Web interfaces is generally considered as desirable and some authors have speculated that because of individual differences among DSS users that a customized interface would improve usability, frequency of use and effectiveness of a DSS [5,41,80]. The controversy about using cognitive style as an individual difference factor in the early 1980s [37] hindered research on these issues. In light of recent technology developments, this stream of research needs to be revisited.

B5. What is the impact of making decisions with a model-driven DSS in a “highly realistic”, simulated decision environment on a person’s actual decision making in the “real” decision environment? How can Visual Interactive Simulation (VIS) be used more effectively to examine the consequences of alternative decision strategies and policies?

The increased capability to develop graphical, immersive, “highly realistic” decision situations creates new challenges and opportunities for users. For example, Pfeil et al. [60] describe an application of a simulation-based DSS for training as well as for design and operation of a plant. According to the authors, the DSS was used at Visteon Automotive System to enhance production efficiency of front axles for several Ford vehicles. They reported productivity improvement of over 30%. Visual Interactive Simulation has primarily been used with Monte Carlo simulation and some technical issues remain, but the impact of this type of simulation deserves more in-

vestigation. When decision makers can observe a realistic simulation it is reasonable to suppose that their perception and understanding of the decision situation will change. Given that most simulations are simplified representations of a situation it is also reasonable to argue that some visual representations may actually hinder or bias decision-making [4]. Predicting and anticipating both positive and negative consequences of specific visual representations is important knowledge for designers. Some research has begun to examine the impact of realistic decision training using decision support on actual decision-making, but much more needs to be done.

4.2. Technical questions

Rapid technology innovation creates new challenges for researchers interested in more technical research questions. Many technical issues related to building and using model-driven DSS have not been resolved. Some technical questions are however, intertwined with behavioral issues. The following questions do not seem adequately addressed in the current literature.

T1. What technology capabilities are needed in the next generation of model-driven DSS generators, especially for creating real-time, model-driven decision support systems? How can “real-time” model-driven DSS be interfaced with “real-time” data-driven DSS to improve decision-making?

In this context, “real-time” refers to a contemporaneous analysis using a model-driven DSS while data about events is being received and displayed. The technologies need to provide speed in model development and updating of data used by the model-driven DSS. This research may involve development of “drag and drop” model components while ensuring semantic and syntactic correctness to build new models for use in rapid development and deployment of DSS. Real-time data collection and storage issues need more investigation, as do technical issues associated with providing various model-driven DSS for use in such an environment [7,8]. The airline industry, for example, faces many problems of making real-time decisions to optimize new routes and rosters after flight cancellations due to weather and other reasons. These problems require massive computational power as well as exten-

sive communications bandwidth and data storage [78]. The same capabilities are also needed for real-time pricing decisions in the optimal demand management applications mentioned in an earlier section.

T2. Is a specific extensible mark-up language (XML) needed for communicating data about model parameters? If so, what mark-up tags can create a core for communicating data to various models?

XML can facilitate interoperability of model-driven DSS developed on different platforms in different model, data and software environments. Some exploratory research has been conducted on creating a decision support mark-up language (DSML). A widely adopted DSML by all industry participants that is implemented in DSS generators can create more powerful Web-based, model-driven DSS. To realize this outcome all parties including vendors and developers must develop and use the standard. However, the varying modeling terminology in use and the variety of categories of DSS suggest that it may be advantageous to have more narrowly defined XML for specific types of models such as optimization or discrete event simulation [25,35]. But there may be advantages to go beyond technique-specific mark-up languages and extend the mark-up to include the dialog and data subsystems of a DSS as well. Development of such standards might permit easier integration of model development and integration of solution techniques from different vendors. Some optimization software vendors such as Maximal Software are working on developing XML specifications for model integration, but no standard has emerged.

Web services are reusable application components that dynamically interact with each other using standard protocols over the Internet. If appropriate XML implementations are created for model-driven DSS, then it is also possible to employ SOAP and UDDI to publicize and reuse model components (be it data, model, solution, or presentation mechanisms) to implement the best model or models. Simple Object Access Protocol (SOAP) is a protocol for exchange of information in a decentralized, distributed environment. It is an industry-standard message format that enables message-based communications for Web services. Universal Discovery Description and Integration (UDDI) is a directory of Web services. The Web Services Description Language (WSDL) is an XML

format for describing a Web Service. Essentially, it describes: the operations (methods) a service provides; details of the data formats and protocols necessary; and the details of protocol-specific network address (URL). These two standards can be used to find a specific modeling or solution capability, and then to invoke it as part of a specific model-driven DSS.

T3. What is an appropriate analytical framework for aggregating results from multiple models or model components? Can Web services provide a technical platform for implementing model aggregation?

Aggregating results depend upon why aggregation is desired and upon what model results seem to warrant aggregation. It has been suggested that Web services provide a means of dynamically aggregating model results when that is appropriate. These capabilities are especially critical for bringing in the “best of breed” solutions to develop, solve, and process a model in a model-driven DSS.

The Application Service Provider (ASP) model that was “hyped” and failed early in the Internet boom is becoming practicable due to the availability of a Web services framework. Issues that need to be resolved include authentication, security, metering the use of model components and creating a mechanism for payment on a per use basis. As these issues are resolved, it seems plausible that the next generation of model-driven DSS will be built using a model development capability from one ASP and the corresponding model solution capability from another ASP. Then the DSS will present the results using the best of the breed visualization and interface capabilities from a third ASP, all in a seamless and transparent manner to the DSS end user.

Integration of model-driven DSS with other DSS development technologies such as expert systems, artificial intelligence models, neural networks, etc. is also an active area of research [33,51,73].

T4. Can the Uniform Modeling Language (UML) help developers and users of model-driven DSS better understand general categories of model-driven DSS applications like resource allocation, sourcing and estimating?

Identifying and modeling processes such as resource allocation or scheduling using UML [10] have been explored, but the possibilities seem

promising and more should be done to specify decision processes where quantitative models might be useful. Some processes that should be better defined include (1) assignment (of tasks, of staff, of resources), (2) capacity planning (also queuing and congestion), (3) estimation of costs, quantities, revenues, (4) evaluation and selection (includes using cost-benefit analysis, financial analysis, multicriteria analysis), (5) location analysis (site selection), (6) routing (vehicles, network packets, people), (7) resource allocation, and (8) sequencing and scheduling [64].

T5. What software capabilities are most appropriate for developing a collaborative, model-driven DSS? Which collaboration technologies are essential and how do they impact information overload?

Collaborative building of model-driven DSS and collaborative use of model-driven DSS are both interesting areas for further research. The “how” of supporting collaboration may be the same or it may differ in these two situations. Besides integrating the use of collaborative communication technologies such as chat, desktop sharing, whiteboarding, voice exchange, video conferencing, etc. into the model development and use processes, advanced interfaces for simultaneous manipulation of models and results need to be explored. Also, it is important to study what kinds of information overload issues occur in this richer media environment.

T6. What are the tradeoffs among various model-driven DSS delivery mechanisms such as a Web browser, spreadsheet, immersive graphics, or peer-to-peer deployment? What innovative user interfaces should be incorporated in next generation model-driven DSS? What data should the user interface software store from its interaction with a regular user?

Designing the user interface has always been of enormous importance in building any type of DSS and new technologies may be especially useful in enhancing the interaction of an end user with a model-driven DSS. Researchers need to examine the value added of developing a model-driven DSS that is able to present its results using multimedia interfaces including synthesized speech. Given that many organizations are multinational and that DSS users have multicultural backgrounds,

model-driven DSS interfaces components need to be studied in diverse decision situations with a wide-range of users. Finally, development of user interfaces personalized to a specific user is now possible because a user’s past interaction with a model-driven DSS can be logged and analyzed, and the subsequent use of the DSS can be adapted to the user’s needs and preferences. Much research needs to be conducted on such adaptive interfaces in the context of model-driven DSS development and usage.

T7. How can developers structure “communities of software agents” that imitate social structures like markets, organizations, or nations to assist decision makers for forecasting and planning? Can multi-agent simulations assist managers in understanding emergent behavior in a particular domain and help predict social trends?

Realistic simulation using multiple agents for decision support [34] is an exciting research frontier that provides many issues for technical research. Until recently, simulation has been primarily used for one-time, special decision support studies rather than as a model-component in a model-driven DSS. This use of simulation techniques can change as new software tools make it easier to create visual simulations [56]. Visual agent-based simulation means managers can see a graphic display of simulation activities, events and results. Will Wright’s games “The Sims”, “SimCoaster” and “SimCity” (<http://thesims.ea.com/>) are the precursors for advanced, agent-based, model-driven DSS. Both technical questions associated with developing the systems and behavioral questions associated with their impact must be investigated.

T8. What reusable model objects should be developed for use in an object-oriented, model-driven DSS development environment?

The range of potential quantitative model components is very large and diverse. We need to investigate what objects will be useful and how they can best be implemented. In general this should be an open source research effort [32]. The model management literature [42] has studied this issue for a long time, but recent advances in software development platforms (Java, VB, C#),

databases, and communications capabilities require a “fresh look” at model management issues.

T9. What extract, transform and load (ETL) capabilities are needed for users of a model-driven DSS? How should the ETL software change depending upon the data, the DSS development environment and the preferences of developer and users?

Extracting, transforming and loading data are part of building data-driven DSS, and occasionally model-driven DSS. In some model-driven DSS, a user may enter all of the data needed by the system. The DSS performs data validation and data storage. The data entry may be 5–15 parameter values, text or other inputs. No data is imported from a source system and no ETL tools are needed. In other model-driven DSS, a time series of data on one or more variables may need to be imported into the DSS. The data set may be 1000 to even 10,000 values. It is often necessary to perform extract and transform tasks to create the data set. Another common type of model-driven DSS uses a small number of data values from an external database that is needed for an analysis by the DSS user. The user defines the analysis and inputs some parameter values. For example, many model-driven investment DSS extract data from a historical stock market database. Finally, some model-driven DSS need very large data sets that the DSS user can interact with. These data sets are created and data may be imported from video files, maps and other sources. As the external data needs of a model or models in a model-driven DSS increases, it becomes more likely that specialized ETL software will be needed to help the DSS developer create the specific decision support data store.

5. Conclusions

Decision support researchers and especially those interested in using quantitative models to build model-driven DSS have an ambitious set of issues that need to be resolved. The behavioral research issues associated with building and using model-driven DSS have often been of relatively low importance because specialists and intermediaries have used the computerized complex models for decision support analyses. That

approach is very limiting and costly. The technical DSS research issues are numerous and equally challenging. Indeed, many DSS research questions have both technical and behavioral dimensions to them that are often inseparable.

Model-driven DSS need to be distributed more widely in organizations and they need to be used by managers and staff for planning and analysis. Model-driven DSS developers have much more to learn about the management of models and there is a need for new development environments to advance the state of the art in building visual interactive DSS. Integrating multiple models in decision support systems should be easier and more common with improved object-oriented DSS development environments. Also, model management in distributed computing environments is now a requirement and not just a possibility and hence this technology needs to be better understood.

Any effort to increase the integration of decision support capabilities in decision processes will benefit from an improved understanding of the various types of decision support systems and especially of model-driven DSS. Identifying specific decision support subsystems assists developers in distributing DSS capabilities and in using components from application service providers. In situations where extensive data sharing is not needed, decision support integration can be achieved using portals rather than by developing a more tightly integrated application. Integration of data- and model-driven DSS presents special challenges in real-time, event-driven situations and a better understanding of the model-driven subsystems should enhance the usability of such systems. Technologies that enable us to develop and deliver a new generation of model-driven DSS are evolving quickly, and the opportunities to study technical and behavioral aspects of such systems are also increasing.

Finally, it is both possible and practical to build and deploy more sophisticated model-driven DSS. Researchers can and should study the enumerated research questions in the context of such systems. Current research issues are challenging, interesting and have practical significance. Future research should lead to improved computerized support for both decision-making and planning. Resolving the behavioral and technical questions identified should

improve and enhance our understanding of DSS and positively impact the performance, acceptance and adoption of next generation model-driven decision support systems. Also, perhaps this identification of issues will stimulate a more coherent stream of research about model-driven DSS.

Acknowledgements

The authors gratefully acknowledge the helpful comments on an earlier version of this paper by Ravindra Ahuja, Sean Eom, Michelle Hanna and Merrill Warkentin. We also acknowledge the helpful feedback from participants in the first SIG DSS workshop held in Seattle, WA in December 2003. Also, some of the material in this article appeared in various issues of DSS News.

References

- [1] M. Abdellaoui, Parameter-free elicitation of utilities and probability weighting functions, *Management Science* 46 (11) (2000 (Nov.)) 1497–1512.
- [2] R.J. Aldag, D.J. Power, An empirical assessment of computer-assisted decision analysis, *Decision Sciences* 17 (4) (1986) 572–588.
- [3] S.L. Alter, *Decision Support Systems: Current Practice and Continuing Challenge*, Addison-Wesley, Reading, MA, 1980.
- [4] P.C. Bell, R. O'Keefe, An experimental investigation into the efficacy of visual interactive simulation, *Management Science* 41 (6) (1995) 1018–1038.
- [5] I. Benbassat, A.S. Dexter, Individual differences in the use of decision support aids, *Journal of Accounting Research* 20 (1) (1982 (Spring)) 1–11.
- [6] P.G. Benson, S.P. Curley, G.F. Smith, Belief assessment: an underdeveloped phase of probability elicitation, *Management Science* 41 (10) (1995) 1639–1653.
- [7] I. Benyahia, J. Potvin, Decision support for vehicle dispatching using genetic programming, *IEEE Transactions on Systems, Man and Cybernetics. Part A. Systems and Humans* 28 (3) (1998 (May)) 306–314.
- [8] G.E.G. Beroggi, W.A. Wallace, The effects of reasoning logics on real-time decision making, *IEEE Transactions on Systems, Man and Cybernetics. Part A. Systems and Humans* 27 (6) (1997 (November)) 743–749.
- [9] R.H. Bonczek, C.W. Holsapple, A.B. Whinston, *Foundations of Decision Support Systems*, Academic Press, New York, 1981.
- [10] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, Boston, MA, 1999.
- [11] E.A. Boyd, J.C. Bilegan, Revenue management and e-commerce, *Management Science* 49 (10) (2003 (October)) 1363–1386.
- [12] M.E. Bratman, D.J. Israel, M.E. Pollack, Plans and resource-bounded practical reasoning, *Computational Intelligence* 4 (1988) 349–355.
- [13] T.A. Bresnick, D.M. Buede, A.A. Pisani, L.L. Smith, B.B. Wood, Airborne and space-borne reconnaissance force mixes: a decision analysis approach, *Military Operations Research* 3 (4) (1997) 65–78.
- [14] E.R. Butchers, P.R. Day, A.P. Goldie, S. Miller, J.A. Meyer, D.M. Ryan, A.C. Scott, C.A. Wallace, Optimized crew scheduling at air New Zealand, *Interfaces* 31 (1) (2001) 30–56.
- [15] R.T. Clemen, R.C. Kwit, The value of decision analysis at Eastman Kodak company, 1990–1999, *Interfaces* 31 (5) (2001) 74–92.
- [16] Decisioneering Staff, SunTrust Banks on Crystal Ball for Assessing the Risk of Commercial Loans, Decisioneering, Inc., <http://dssresources.com> (2001).
- [17] D. Delen and R. Sharda, Forecasting Box-office Receipts of Motion Pictures Using Neural Networks, Working Paper, Oklahoma State University, 2004.
- [18] G. DeSanctis, G. DeSanctis, R. Brent, A foundation for the study of group decision support systems, *Management Science* 33 (5) (1987) 589–609.
- [19] F.N. de Silva, R.W. Eglese, Integrating simulation modeling and GIS: spatial decision support systems for evacuation planning, *Journal of the Operational Research Society* 51 (4) (2000) 423–430.
- [20] M. d'Inverno, D. Kinny, M. Luck, M. Wooldridge, A formal specification of dMARS, in: M. Singh, A. Rao, M. Wooldridge (Eds.), *Intelligent Agents IV: Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages*, Springer-Verlag, 1998, pp. 155–176 (LNAI 1365).
- [21] D.J. Dunning, S. Lockfort, Q.E. Ross, P.C. Beccue, J.S. Stonebraker, New York power authority uses decision analysis to schedule refueling of its Indian point 3 nuclear power plant, *Interfaces* 31 (5) (2001) 121–135.
- [22] S.B. Eom, *Decision Support Systems Research (1970–1999) as Cumulative Tradition and Reference Disciplines*, Edwin Mellen Press, Lewiston, NY, 2002.
- [23] S.B. Eom, *Decision Support Systems Research and Reference Disciplines (1970–2001): A Research Guide to the Literature and an Unobtrusive Bibliography with Citation Frequency*, Edwin Mellen Press, Lewiston, NY, 2003.
- [24] J.R. Evans, D.L. Olson, *Introduction to Simulation and Risk Analysis*, 2nd edition, Prentice Hall, Upper Saddle River, NJ, 2002.
- [25] O.C. Ezechukwu and I. Maros, ORML: Optimization Reporting Markup Language, Department of Computing Technical Report 2003/13, Imperial College London, <http://www.doc.ic.ac.uk/research/technicalreports/2003> (November, 2003).
- [26] C. Fierbinteanu, A decision support systems generator for transportation demand forecasting implemented by constraint logic programming, *Decision Support Systems* 26 (3) (1999) 179–194.

- [27] L.R. Foulds, D.G. Johnson, SlotManager: a microcomputer-based decision support system for university timetabling, *Decision Support Systems* 27 (4) (2000) 367–381.
- [28] R. Fourer, 2003, Software Survey: Linear Programming, *OR/MS Today* 30, No. 6 (December 2003) 34–35, 42–43.
- [29] R. Gimblett, B. Durnota and R. Itami, Spatially-explicit Autonomous Agents for Modeling Recreation Use in Complex Wilderness Landscapes, *Complexity International* 3 (1996), <http://journal-ci.csse.monash.edu.au/ci/vol03/>.
- [30] S. Guerlain, G.A. Jamieson, P. Bullemer, R. Blair, The MPC elucidator: a case study in the design for human–automation interaction, *IEEE Transactions on Systems, Man and Cybernetics. Part A. Systems and Humans* 32 (1) (2002) 25–39.
- [31] H.B. Hayes, *Decision Support Systems*, Washington Technology 16, No. 13 (2001), <http://www.washingtontechnology.com/>.
- [32] P. Heilman, G. Davis, P. Lawrence, J.L. Hatfield and J. Huddleston, The Facilitator—An Open Source Effort to Support Multiobjective Decision Making, *The International Environmental Modeling and Software Society 2002 Online Proceedings*, <http://www.iemss.org/>.
- [33] M. Henrion, J.S. Breese, E.J. Horvitz, Decision analysis and expert systems, *AI Magazine* 12 (4) (1991) 64–91.
- [34] T.J. Hess, L.P. Rees, T.R. Rakes, Using autonomous software agents to create the next generation of decision support systems, *Decision Sciences* 31 (1) (2000) 1–31.
- [35] R.L. Hobbs, Using XML to Support Military Decision-making, <http://idealliance.org/> (December 2003).
- [36] S.J. Hoch, D.A. Schkade, A psychological approach to decision support systems, *Management Science* 42 (1) (1996) 51–64.
- [37] G.P. Huber, Cognitive style as a basis for MIS and DSS design: much ado about nothing? *Management Science* 29 (1983) 567–579.
- [38] E. Katok, D. Ott, Using mixed-integer programming to reduce label changes in the Coors aluminum can plant, *Interfaces* 30 (2) (2000) 1–12.
- [39] D. Keefer, C. Kirkwood, J. Corner, Perspective on decision analysis applications, 1990–2001, *Decision Analysis*, Preview Issue, 2003 (October), pp. 5–24.
- [40] E. Kim and S.B. Eom, *Decision Support Systems Applications Research: A Bibliography (1995–2001)*, Working Paper, Southern Missouri State University, 2004.
- [41] J.E. Kottemann, W.E. Remus, A study of the relationship between decision model naturalness and performance, *MIS Quarterly* 13 (2) (1989) 171–181.
- [42] R. Krishnan, K. Chari, Model management: survey, future research directions and a bibliography, *Interactive Transactions of OR/MS* 3 (1) (2000).
- [43] J. Kuljis, R.J. Paul, An appraisal of web-based simulation: whither we wander? *Simulation Practice and Theory* 9 (2001) 37–54.
- [44] O.I. Larichev, D. Yu, L.L. Ustinovicus, Multicriteria method for choosing the best object for investments, in: T. Bui, H. Sroka, S. Stanek, J. Goluchowski (Eds.), *DSS in the Uncertainty of the Internet Age*, Karol Adamiecki University of Economics in Katowice, Katowice, PL, 2003, pp. 255–270.
- [45] L.J. LeBlanc, D. Randels Jr., T.K. Swann, Heery international's spreadsheet optimization model for assigning managers to construction projects, *Interfaces* 30 (6) (2000) 95–116.
- [46] H.L. Lee, S. Whang, Demand chain excellence: a tale of two retailers, *Supply Chain Management Review* 5 (3) (2001) 40–46.
- [47] T. Lee, J. Kao, C. Wu, Application of PDSS to improve the pricing efficiency of wholesale fish markets, *Simulation Practice and Theory* 9 (2002) 241–253.
- [48] Manugistics Demand and Revenue Management (DRM) Solutions, <http://www.manugistics.com> (June 2004).
- [49] D.L. Margulius, Dawn of the Real-time Enterprise, *InfoWorld*, <http://www.infoworld.com> (January 17, 2002).
- [50] J. Marsan, Smarter Revenue Management Systems, *Hotels Magazine*, <http://www.hotelsmag.com/0399/0399tech.html> (March, 1999).
- [51] I. Matzkevich, B. Abramson, Decision analytic networks in artificial intelligence, *Management Science* 41 (1995) 1–22.
- [52] D.T. Maxwell, Decision analysis: aiding insight VI—it's not your grandfather's decision analysis software, *OR/MS Today* 29 (3) (2002) 44–51.
- [53] A. McClain, Managing Consumer Demands: Art Becoming Science, *Retailwire* (March 4, 2004).
- [54] J. Miller, P.A. Fishwick, S.J.E. Taylor, P. Benjamin, B. Szymanski, Research and commercial opportunities in web-based simulation, *Simulation Practice and Theory* 9 (2001) 55–72.
- [55] M. Mollaghasemi, J. Pet-Edwards, *Technical Briefing: Making Multiple-Objective Decisions*, IEEE Computer Society Press, Los Alamitos, CA, 1997.
- [56] C.A. Murphy, T. Perera, The definition of simulation and its role within an aerospace company, *Simulation Practice and Theory* 9 (2002) 273–291.
- [57] J.F. Nunamaker, L.M. Applegate, B.R. Konsynski, Computer-aided deliberation: model management and group decision support, *Operations Research* 36 (6) (1988 (Nov./Dec.)) 826–848.
- [58] Optims, <http://optims.com> (June, 2004).
- [59] OPUS2 TopLine PROPHET Yield Management and Forecasting Solutions, <http://www.opus2.com> (2004, June).
- [60] G. Pfeil, R. Holcomb, C.T. Muir, S. Taj, Visteon's sterling plant uses simulation-based decision support in training, *Operations and Planning*, *Interfaces* 30 (1) (2000) 115–133.
- [61] V.V. Podinovski, A DSS for multiple criteria decision analysis with imprecisely specified trade-offs, *European Journal of Operational Research* 113 (2) (1999) 261–270.
- [62] D.J. Power, *Decision Support Systems Hyperbook*, <http://dssresources.com/dssbook/> (Fall, 2000).
- [63] D.J. Power, A history of microcomputer spreadsheets, *Communications of the Association for Information Systems* 4 (9) (2000 (October)) 154–162.
- [64] D.J. Power, *Decision Support Systems: Concepts and Resources for Managers*, Greenwood/Quorum Books, Westport, CT, 2002.
- [65] D.J. Power, *Free Decision Support Systems Glossary*, <http://dssresources.com/glossary/> (2004).
- [66] D.J. Power, Specifying an expanded framework for classifying and describing decision support systems, *Communica-*

- tions of the Association for Information Systems 13 (13) (2004) 158–166.
- [67] D.J. Power, S. Meyeraan, R. Aldag, Impacts of problem structure and computerized decision aids on decision attitudes and behaviors, *Information and Management* 26 (1994) 281–294.
- [68] C. Ragsdale, *Spreadsheet Modeling and Decision Analysis*, South-Western Thomson Learning, Cincinnati, OH, 2000.
- [69] C. Ragsdale, D.J. Power, P.K. Bergey, Spreadsheet-based DSS curriculum issues, *Communications of the Association for Information Systems* 9 (21) (2002 (November)) 356–365.
- [70] J. Rothfeder, Expert Voices: Icosystem's Eric Bonabeau, CIOInsight.com, <http://www.cioinsight.com/article2/0,3959,1124316,00.asp> (June 16, 2003).
- [71] R. Sharda, S. Barr, J. McDonnell, Decision support systems effectiveness: a review and an empirical test, *Management Science* 34 (2) (1988) 139–159.
- [72] J.P. Shim, M. Warkentin, J.F. Courtney, D.J. Power, R. Sharda, C. Carlsson, Past, present, and future of decision support technology, *Decision Support Systems*, Special Issue, DSS: Directions for the Next Decade 33 (2) (2002) 111–126.
- [73] B.G. Silverman, Unifying expert systems and the decision sciences, *Operations Research* 42 (1994) 393–413.
- [74] B.C. Smith, J.F. Leimkuhler, R.M. Darrow, Yield management at American airlines, *Interfaces* 22 (1) (1992) 8–31.
- [75] S. Smith, N. Agrawal, A. Tsay, A decision support system for retail supply chain planning for private-label merchandise with multiple vendors, in: J. Geunes, P. Pardalos, H. Romeijn (Eds.), *Supply Chain Management Model, Applications, and Research Directions*, Kluwer, 2003, pp. 163–198.
- [76] R.H. Sprague Jr., A framework for the development of decision support systems, *Management Information Systems Quarterly* 4 (4) (1980) 1–26.
- [77] R.H. Sprague, E.D. Carlson, *Building Effective Decision Support Systems*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1982.
- [78] J.A. Stankovic, S.H. Son and J. Hansson, Misconceptions About Real-time Databases, *Computer* (June 1999) 29–36, <http://www.cs.virginia.edu/~stankovic/psfiles/R6stan.lo.pdf>.
- [79] J. Swain, Simulation reloaded, *OR/MS Today* 30 (4) (2003).
- [80] M. Swink, The influences of user characteristics on performance in a logistics DSS application, *Decision Sciences* 26 (4) (1995) 503–530.
- [81] F. Ülengin, Y.I. Topcu, S.Ö. Sahin, An integrated decision aid system for Bosphorus water-crossing problem, *European Journal of Operational Research* 134 (1) (2001) 179–192.
- [82] C.P.B. van der Ploeg, C.V. Vliet, S.J. De Vlas, J.O. Ndinya-Achola, L. Fransen, G.J. Oortmarssen, J.D.F. Habbema, STDSIM: a microsimulation model for decision support in sexually transmitted diseases control, *Interfaces* 28 (3) (1998) 84–100.
- [83] B. Vigus, G.G. Brown, J. Keegan, K. Wood, The Kellogg company optimizes production, *Inventory and Distribution*, *Interfaces* 31 (6) (2001) 1–15.
- [84] D. von Winterfeldt, W. Edwards, *Decision Analysis and Behavioral Research*, Cambridge University Press, Cambridge, UK, 1986.
- [85] O.J. Wasyluk, D. Saaty, Developing a Portfolio Approach to Capital Investment: A Case Study in Re-engineering Resource Allocation at the U.S. Department of Veteran's Affairs, Expert Choice, Inc., <http://dssresources.com> (2001).
- [86] K.P. Yoon, C. Hwang, *Multiple Attribute Decision Making: An Introduction*, Sage Publications, Thousand Oaks, CA, 1995.



Daniel J. Power is a Professor of Information Systems and Management at the College of Business Administration at the University of Northern Iowa, Cedar Falls, Iowa and the editor of *DSSResources.COM*, the Web-based knowledge repository about computerized systems that support decision making, the editor of *PlanningSkills.COM*, and the editor of *DSS News*, a bi-weekly e-newsletter. Dan writes the column "Ask Dan!" in *DSS News*.

Dr. Power's research interests include the design and development of Decision Support Systems and how DSS impact individual and organizational decision behavior.

Since 1982, Dan Power has published more than 40 articles, book chapters and proceedings papers. His articles have appeared in leading journals including *Decision Sciences*, *Decision Support Systems*, *Journal of Decision Systems*, *MIS Quarterly*, *Academy of Management Review*, and *Information and Management*. He is also co-author of a book titled *Strategic Management Skills* and he authored a book on *Decision Support Systems*. His DSS book (2002) is a broad ranging handbook on the fundamentals of building decision support systems. His expanded DSS Framework has received widespread interest.

Professor Power is a member of three academic journal editorial boards, the section editor of the ISWorld pages on Decision Support Systems and founding Chair of the Association for Information Systems Special Interest Group on Decision Support, Knowledge and Data Management Systems (SIG DSS).

In 1982, Professor Power received a Ph.D. in Business Administration from the University of Wisconsin-Madison. He was on the faculty at the University of Maryland-College Park from 1982 to 1989 and received tenure as an Associate Professor in 1987. Power served as the Head of the Management Department at the University of Northern Iowa (UNI) from August 1989 to January 1996. He served as Acting Dean of the UNI College of Business Administration from January 16, 1996 to July 31, 1996. Dr. Power has been a visiting lecturer at universities in China, Denmark, Ireland, Israel, and Russia. Power has consulted with a number of organizations and in Summer 2003 he was a Visiting Faculty Research Fellow with the U.S. Air Force Research Lab Information Directorate (AFRL/IF).

Dr. Power is a pioneer developer of computerized decision aiding and support systems. During 1975–1977, he developed a computerized system called DECAID, DECision AID. In 1981–1983, he reprogrammed and expanded the system for the Apple II PC. In 1986–1987, he designed a set of decision aiding tools for the Management Decision Assistant package from Southwestern Publishing.



Ramesh Sharda is Regents Professor of Management Science and Information Systems and Conoco/DuPont Chair of Management of Technology at Oklahoma State University. He is the Vice Chair and Chair-elect of AIS SIG DSS. He served as Secretary/Treasurer of SIG DSS from 2002 to 2004.

He received his B.Eng. degree from University of Udaipur, M.S. from The Ohio State University, and an MBA and Ph.D.

from the University of Wisconsin-Madison. One of his major activities in the last few years was to start the MS in Telecommunications Management Program at Oklahoma State University. He has served as the founding editor of *Interactive Transactions of OR/MS*, an INFORMS electronic journal. He is also the computer science editor of *OR/MS Today*, and an associate editor of the *INFORMS Journal on Computing*. Ramesh has co-edited several books and is the editor for a Kluwer book series in Computer Science/Operations Research Interfaces, which has currently published 17 volumes. His research has been published in major journals in management science and information systems including *Management Science*, *Information Systems Research*, *Decision Support Systems*, *Interfaces*, *INFORMS Journal on Computing*, *Computers and Operations Research*, and many others. His research interests are in optimization applications on desktop computers, information systems support for new product development, neural networks, business uses of the Internet, and knowledge networks. His research has been sponsored by the Defense Logistics Agency, NSF, Marketing Science Institute, and several other organizations. He and his colleagues are working on using information technology to facilitate electronic commerce between the US government and small businesses. This work, sponsored by the CATT program, has resulted in development of SCORE, a tutorial for small businesses, hands on Internet training materials, and development of a web site, as well as the development of the Defense Supplier Catalog. Ramesh is also a cofounder of a company that produces virtual trade fairs, iTradeFair.com.