Chapter VIII
Design and Analysis of Decision Support Systems

John Wang
Montclair State University, USA

James Yao
Montclair State University, USA

Qiyang Chen
Montclair State University, USA

Ruben Xing
Montclair State University, USA

ABSTRACT

Since their creation in the early 1960's, decision support systems (DSSs) have evolved over the past 4 decades and continue to do so today. Although DSSs have grown substantially since its inception, improvements still need to be made. New technology has emerged and will continue to do so and, consequently, DSSs need to keep pace with it. Also, knowledge needs to play a bigger role in the form of decision making. We first discuss design and analysis methods/techniques/issues related to DSSs. Then, the three possible ways to enhance DSSs will be explored.

INTRODUCTION

Over the 4 decades of its history, decision support systems (DSSs) have moved from a radical movement that changed the way information systems were perceived in business, to a mainstream commercial information technology movement that all organizations engage. This interactive, flexible, and adaptable computer based information system derives from two main areas of research: the theoretical studies of organizational decision making done at the Carnegie Institute in the 1950's and early 1960's, as well as the technical work on interactive computer systems which was mainly performed by the Massachusetts Institute of Technology (Keen & Morton, 1978).

DSSs began due to the importance of formalizing a record of ideas, people, systems, and technologies implicated in this sector of applied information technology. But the history of this system is not precise due to the many individuals involved in different stages of DSSs and various industries while claiming to be pioneers of the system (Power, 2003; Arnott & Pervan, 2005). According to Arnott (2006), the DSS field began in the early 1970s as a radical alternative to large-scale management information systems (MIS). Over time, major changes in information technology have enabled new decision support movements. In the late 1980s and mid-1990s, multidimensional modeling, OLAP technology, and advances in storage technology and data modeling led to the deployment of large-scale executive information systems, data warehousing, and business intelligence. Now DSSs have become very sophisticated and stylish since the early pioneering research. Many new systems have expanded the frontiers established by these pioneers, yet the core and basis of the system remains the same. Today, DSSs are used in the finance, accounting, marketing, medical, as well as many other fields.

BACKGROUND

The basic ingredients of a DSS can be stated as follows: the data management system, the model management system, the knowledge engine, the user interface, and the users (Donciulescu, Filip, & Filip, 2002). The database is a collection of current or historical data from a number of application groups. Databases can range in size from storing it in a PC that contains corporate data that has been downloaded, to a massive data warehouse that is continuously updated by major organizational transaction processing systems (TPSs). When referring to the model management system, it is primarily a stand-alone system that uses some type of model to perform "what if" and other kinds of analysis. This model must be easy to use, and therefore the design of such model is based on a strong theory or model combined with a good user interface.

A major component of a DSS is the knowledge engine. To develop an expert system requires input from one or more experts, this is where the knowledge engineers go to work, who can translate the knowledge as described by the expert into a set of rules. A knowledge engineer acts like a system analyst but has special expertise in eliciting information and expertise from other professionals (Laudon & Laudon, 2005).

The user interface is the part of the information system through which the end user interacts with the system, the type of hardware, and the series of on-screen command and responses required for a user to work with the system. An information system will be considered a failure if its design is not compatible with the structure, culture, and goals of the organization. Research must be conducted to design a close organizational fit, to create comfort and reliability between the system and user. In a DSS, the user is as much a part of the system as the hardware and software. The user can also take many roles such as decision maker, intermediary, maintainer, operator, and feeder. A DSS may be the best one in its industry but it still requires a user to make the final decision.

Power (2003) introduced a conceptual level of DSSs, which contains five different categories. These categories include model-driven DSS, communication-driven DSS, data-driven DSS, document-driven DSS, and knowledge-driven DSS. Defining DSS is not always an easy task due to the many definitions available. Much of this problem is attributed to the different ways a DSS can be classified. At the user level, a DSS can be classified as passive, active, or cooperative.

Essentially, DSS is a computer-based system that provides help in the decision making process. However, this is a broad way of defining the subject. A better way of describing DSS is to say it is a flexible and interactive computer-based system that is developed for solving non-structured management problems. Basically, the system uses information inputted from the decision maker (data and parameters) to produce an output from the model that ultimately assists the decision maker in analyzing a situation. In the following sections, we first discuss design and analysis methods/techniques/issues related to DSSs. Then, the three possible ways to enhance DSSs will be explored. At the end, future trends in DSSs will be discussed.

DESIGN AND ANALYSIS METHODS/ TECHNIQUES/ISSUES RELATED TO DSSs

Design Methods

Today, DSSs hold a primary position in an organization's decision making by providing timely and relevant information to decision makers. It has become a key to the success or survival of many organizations. However, there is a high tally of failure in information systems development projects, even though they are a focal point of industrial concern (Goepp, Kiefer, & Geiskopf, 2006). Designing methods have become an important component that assures a successful information system design. This issue is in relevance to the design of a DSS.

There have been many different strategies employed for the design of a DSS, including the early decision-oriented design approach (Stabell, 1983). Current research on DDS design has witnessed the rapid expanding of object-oriented (OO) approach, which exploits object-oriented software engineering with unified modeling language (UML); knowledge management (KM) approach, which supports end-users by embedding declarative and/or procedural knowledge in software agents; structured modeling (SM) approach, which employs a hierarchically organized, partitioned, and attributed acyclic graph to represent models; and design science (DS) approach, which attempts to create artifacts that serve human purposes and solve organizational problems.

Object-oriented (OO) approach: The characteristic of OO approach is to use object-oriented software engineering with UML in the design and implementation of a DSS. OO design concepts are based on software engineering in that knowledge encapsulation present in a set of objects in an object-oriented system, where sub-classes show inheritance of the properties of the main class, is more compact and yet extensive compared to a logic-based system. Higher order logic is required to duplicate the performance of a simple object-oriented system (Pillutla & Nag, 1996). OO approach is considered a novel way of systems thinking. It provides designers and developers with easy analysis of complex systems and design of suitable software systems. It allows the developers and users to think in terms of objects and their behaviors instead of thinking about processes and process complexities. The main advantages are due to the features like data abstraction, encapsulation, inheritance, and polymorphism of OO approach (Nagarur & Kaewplan, 1999).

OO approach involves basically three major steps (Tian, Ma, Liang, Kwok, & Liu, 2005). The user's requirements are first captured by using a set of use case diagrams. These diagrams indicate all the functionalities of the system from the user's point of view. Then classes and their relationships are identified and described in class diagrams. Finally, sequence diagrams or collaboration diagrams are developed, which describe the interaction between objects (instances of classes). Tian et al. (2005) designed a DSS with the OO approach for an organization, which was implemented successfully.

Knowledge management (KM) approach: In some environments (non-preprogrammed applications), end-users, especially the less experienced end-users, need to have certain knowledge guiding them how to use the system. The KM design approach supports end-users by embedding declarative and/or procedural knowledge in software agents.

West Jr. and Hess (2002) used KM design approach to support end-users with spatially oriented decision-making by reporting on a specific spatial DSS that uses the approach. Procedural knowledge about performing spatial analysis was embedded in software agents that assist users with difficult and spatially oriented tasks. Because the system used a metadata repository, users were supported in some tasks by using the metadata repository directly and in others by using the software agents that access the metadata. According to West Jr. and Hess (2002), KM design approach provides better assistance to inexperienced users of spatial DSS, which requires a design approach that will prioritize knowledge support of the end-users' decision-making activities, and this approach, with the distribution of knowledge between metadata, agents, and end-users, has similar potential as a determinant of system success in any DSS (spatial or otherwise) where both declarative and procedural knowledge are needed to effectively accomplish the decision-making task. .

Structured modeling (SM) approach: SM (Geoffrion, 1987) approach "uses a hierarchically organized, partitioned, and attributed acyclic graph to represent models" (Srinivasan & Sundaram, 2000). It is a formal framework for describing models. SM identifies the basic components of models, the relationships among these components, and conditions under which a model may be termed structured (Lenard, 1993). It includes a language for describing a model schema and prescribes data tables for capturing the details of model instances.

SM decomposes a decision problem into genera and elements within genera in a hierarchic way, which forms a system that is sensitive to natural definitions of entities and objects in the problem (Pillutla & Nag, 1996). It consists of three levels: elemental structure, generic structure, and modular structure. The elemental structure intends to capture the definitional detail of a specific model instance. The generic structure targets at capturing the natural familial groupings of elements. The modular structure seeks to organize generic structure hierarchically according to commonality or semantic relatedness. The leveled structures allow the complexity of a model to be managed and ranked according to its hierarchies. The graph feature allows modelers and decision makers to understand the model better. A key advantage of SM is the ease with which structured models can be visualized (Srinivansan & Sundaram, 2000).

Srinivansan and Sundaram (2002) propose using SM in the design of model based DSS with the intention of solving existing design problems, such as lack of theory or design principles and domain specific issues, in other approaches. They select object relational database environment to implement SM, to be specific, they propose using SM to provide a systematic general framework for conceptual modeling and an object relational database management system (ORDBMS) to implement it. They believe an object relational platform for implementing structured models offers a development platform that is well suited to their needs. Such a platform is uniquely capable of meeting the conceptual requirements outlined by SM while satisfying many practical design concerns such as performance, persistence, and interoperability. They trust that their proposition of using SM approach for specific problem conceptualization and such a powerful environment as ORDBMS to implement it will provide design ideas that can potentially serve a very useful class of applications (Srinivansan & Sundaram, 2000).

Design science (DS) approach: Arnott (2006) defines design science as an alternative, or complement, to the natural science approach which has been a dominant research methodology in information systems field. Natural science tries to understand reality, but design science attempts to create things that serve human purposes (March & Smith, 1995). In design science, researchers create and evaluate information technology artifacts that are intended to solve identified organizational problems (Hevner et al., 2004). Design science is especially relevant to information system research because it helps to address the role of the information technology artifact in information system research and the low professional relevance of many information system studies (Benhasat & Zmud, 1999; Orlikowski & Iacono, 2001).

The functionality of a DSS evolves over a series of development cycles where both the end-users and the systems analyst are active contributors to the shape, nature, and logic of the system (Arnott, 2004). Yet system developers have little guidance about how to proceed with evolutionary DSS development.

DSS developers are facing the fact that insufficient knowledge exists for design purpose, and designers must rely on intuition, experience, and trial-and-error methods. Design science approach, on the other hand, can facilitate developers to create and evaluate information technology artifacts that are intended to solve identified organizational problems (Hevner, March, Park, & Ram, 2004). Vaishnavi and Kuechler (as in Arnott, 2006) proposed a design science methodology with the major process steps of awareness of problem, suggestion, development, evaluation, and conclusion. Arnott (2006) proposed a five steps approach, which was adapted from Vaishnavi and Kuechler, for designing evolutionary DSS: problem recognition, suggestion, artifact development, evaluation, and reflection. A research project by Arnott indicates that design science approach can tackle problems of both theoretical and practical importance.

DSS design model: DSS design model is the most recent DSS design approach developed by Klashner and Sabet and is worthy of a discussion. According to Klashner and Sabet (2007), DSS design model differs from any other models and approaches in that it is a more comprehensive design approach to address domain and nondeterministic complexities arising from real-world decision-making requirements. Incorporating morphogenetic principles, the model reflectively and concurrently informs its own evolution and directly impacts the design of the proposed DSS under development "although this new DSS design model appears simple and straightforward" (Klashner & Sabet, 2007).

The model consists of three major components: theory and analysis, simulation, and decision/design. Within the theory and analysis component, multiple data from system domain are fed into theory and analysis. A relation between the theory and domain is maintained to continuously exchange synchronous data and update the theory. The simulation component interacts with the theoretical components to integrate the data feed. As to the decision/design component, the design decision-making process will inevitably be influenced by the effect of theoretical analysis and simulation combined to the degree of not violating the decision-makers' shared understanding of the design goals. Then the newly integrated design decisions immediately act on the relationship between the system domain and information infrastructure, thus completing the first full iteration since the general iterative flow of data.

Klashner and Sabet (2007) argue that in the early years of DSS research, design choices were intuitively understood in most cases because of the straightforward nature of the stakeholder's requirements. But today, because stakeholder decisions have become highly subjective and complicated due to the increased problem complexity arising from various semi- to ill-structured problems (Nemati, Steiger, Iyer, & Herschel, 2002), a more comprehensive systems design approach is needed. Thus, the new DSS design model has been developed to address this issue. At present, the model application is domain specific, but it is currently being applied to another mission critical infrastructure design effort to test its generalizability to other domains where DSS plays a key role in daily operations (Klashner & Sabet, 2007).

Design Techniques

As we are advancing in information technologies, business decision makers can now have access to vast amount of information. On one hand, they may gain necessary and important information for making informed decisions, but on the other hand, they may also become overloaded by the information irrelevant to what they need. Thus, there is a pressing need for decision aiding tools that would effectively process, filter, and deliver the right information to the decision makers. Proper combination of DSSs and agent technologies could prove to be a very powerful tool for rendering decision support (Vahidov & Fazlollahi, Winter 2003-2004).

A software agent performs interactive tasks between the user and the system. The user instructs the system what he/she intends to accomplish. The software agent carries out the task. By analogy, a software agent mimics the role of an intelligent, dedicated and competent personal assistant in completing the user's tasks (Bui & Lee, 1999). In the DSS environment, software agents have been more formally described as autonomous software implementations of a task or goal that work independently, on behalf of the user or another agent (Hess, Rees, & Rakes, 2000). As the traditional, direct manipulation interface of our computing environment is much limited (Maes, 1994), software agents would seem to be a suitable and most needed solution for providing procedural assistance to end-users (West Jr. & Hess, 2002). "These 'robots of cyberspace' can be effectively utilized in automating many information processing tasks" (Vahidov & Fazlollahi, 2003-2004).

In some DSS environments, such as spatial DSS (Sikder & Gangopadhyay, 2002; West Jr. & Hess, 2002), Internet-based DSS (Bui & Lee, 1999) and Web DSS (Vahidov & Fazlollahi, 2003-2004), a multi-agent system should be designed and implemented in the DSS to facilitate the decision makers since decision making involves complex set of tasks that requires integration of supporting agents (Bui & Lee, 1999), and these agents should have behaviors to work in team (Norman & Long, 1994). Vahidov and Fazlollahi (2003-2004) developed architecture of multi-agent DSS for e-commerce (MADEC), in which intelligence team (agents), design team (agents), and choice team (agents) were composed. The multi-agent system was implemented in a prototype of MADEC, which received higher user satisfaction.

THREE POSSIBLE WAYS TO ENHANCE DSSs

Creating Knowledge Warehouses (KW)

Nemati (2002) proposed that a new generation of knowledge-enabled systems that provides the infrastructure required to capture, enhance, store, organize, leverage, analyze, and disseminate not only data and information but also knowledge (Nemati, 2002). Expanding data warehouses to encompass the knowledge needed in the decision making process is the creation of knowledge warehouses (KW). An important component of KW is a very complex process known as knowledge management. Knowledge management allows for knowledge to be converted from tacit to explicit through such processes as filtering, storing, retrieving, and so forth, thus allowing it to be utilized by decision makers.

The goal of KW is to give the decision maker an intelligent analysis standpoint that enhances all aspects of the knowledge management process. The main drawbacks of KW are the amount of time and money that need to be invested as well as some of the same problems that are found in successfully implementing DSSs. Among these factors are the users' involvement and participation, values and ethics, organization and political issues within the company, and other external issues. The development and implementation of KW still has much work to be done, however, DSSs seem to be headed toward knowledge enhancement in the future and KW looks to have a promising outlook in the upcoming years as a result.

Focusing on Decision Support

While knowledge management systems seem like a logical way to advance the shortcomings of DSSs, another view also exists. By removing the word "system" from DSSs and focusing on decision support, decision making might cause some interesting, new directions for research and practice. Decision support (DS) is the use of any plausible computerized or non-computerized means for improving sense making and/or decision making in a particular repetitive or non-repetitive business situation in a particular organization (Alter, 2004).

DS embodies a broader perspective that seems logical in environments where the user does not necessarily need the technical aspects of DSSs. This is based on the belief that most work systems of any significance include some form of computerized support for sense making and decision making (Alter, 2004). The difference between DSSs and DS is not too drastic but DS is a sensible option for many companies due to the increase in technology since the creation of DSSs; DSSs may not fit the needs of a business as it had in the past.

Integrating DSSs and KMSs

In line with Bolloju (2002), integrating decision support and knowledge management may correct some of the deficiencies of DSSs. The decision-making process itself results in improved understanding of the problem and the process, and generates new knowledge. In other words, the decision-making and knowledge creation processes are interdependent. By integrating the two processes, the potential benefits that can be reaped make the concept seem more worthwhile.

Integrating DSSs and KMSs seems to be the best choice out of the three possible ways to enhance DSS. The reasoning behind this selection is that integrating the two seems to provide a way for including both options without sacrificing one for the other. More importantly, while KW appears to have a very bright future, KW currently requires a great amount of time and money. The combination of both areas allows for a better overall utilization in the present. In time, KW may not be as time consuming and costly as it is now. However, to achieve a better balance of usefulness and efficiency, the integration of DSSs and KMSs appears to be the smartest choice.

FUTURE TRENDS

DSS in Business Analytics

The future of DSSs, Angus (2003) argued and supported by SAS (2004), is in the field of *business analytics* (BAs). BAs differ from that of the recently and previously more common business intelligence (BI). With the fast pace of business and life today it would only make sense for a shift to BA because it does focus on the many possibilities and the future outcomes for production and service.

BAs focus on the future of operations. Opposed to that of BI where it focuses on the past and what can be done to change the past if things were done wrong or repeat if things were done right. However, BAs let managers center on what future trends are developing, which allows them not to accumulate a surplus of inventory of outdated products. It also enables managers to change their prices before the market does, or introduce their new product before anyone else gets the chance to. This is known as first-to-market (Gnatovich, 2006). BAs give the companies that use it a tremendous advantage over their competitors in the market place.

Power-Hungry DSS

As everyone can see, the computing power is still accelerating. With its plan of "Itanium" processors, Intel is rushing towards its upcoming generation of 64 bit chips to support power-craving applications. Without any doubt, the power-hungry, large scale, integrated DSS application with dynamic calculation, background data mining, and high-end data visualization falls right into this end of the spectrum (Thomsen, 2003).

Web-Based DSS

The fast growing Internet and e-commerce have greatly impacted the way we conduct business. With the support of intelligent agents, DSS has been implemented for aiding decision makers in e-commerce. E-commerce combines transaction and decision support within an e-framework. Ultimately, the e-world is just another channel (Thomsen, 2003). If we can use the e-channels appropriately, integrate text engines within an overall DSS architecture, and provide for interoperability between DSS components, we may actually learn a few things about all this data we are collecting and make better decisions—without losing our humanity (Thomsen, 2003).

DSS in Retailing Business

In the retailing business sector, the competition over customers grows fierce. A customer's decision of buying from one retailer over another becomes the dividing factor in competition. Additionally, product mix between general and specialty retail is becoming homogenized. This expands the growth for more sophisticated and accurate decision support technology in the business (Rowen, 2005).

A few other trends in the retail business also call our attention. According to Rowen (2005), decision support solutions must interact with forecasting, allocation, and many other demand-chain applications from disparate vendors to increase margins. Additionally, we are being confronted with the coming onslaught of customer, events, and transactional data. Retailers are generating information at a volume that few can find significant use for today. Finally, forecasting and tracking anticipated sales has become another decision support trend within retail. Such technology enables a retailer to investigate the situation at hand, determining if a product is available.

Research in DSS

In terms of trends in DSS research, in the late 1980s and early 1990s, data warehouse and data mining was one of the focuses of research in DSS field. Since then software agents, also know as intelligent agents, emerged as an interdisciplinary area involving researchers from such fields as expert systems, DSS, cognitive science, psychology, databases, and so forth (Eom, 1999). Citations relevant to software agents in this chapter support such a proposition.

Studies of design science have shown a significant number in recent years in DSS research (Arnott & Pervan, 2005). This research stream has been rapidly expanding with a range of new topics encouraged by technological advances, methodological innovations, and increased expectations for theory development and empirical analysis of the new artifacts (Banker & Kauffman, 2004), such as the design of auction mechanisms (Bapna, Goes, & Gupta, 2003; Kelly & Steinberg, 2000) and optimal strategies for investment in knowledge by using a market mechanism (Ba, Stallaert, & Whinstone, 2001). It is likely that it will remain as a major topic in years to come.

According to Eom's (Eom, 1999) study, Web-based DSS is another emerging topic in the DSS area. Sikder and Gangopadhyay (2002) identified research issues on the design and implementation of a Web-base collaborative spatial decision support system. Vahidov and Fazlollahi (2003-2004) proposed architecture for a multi-agent DSS for e-commerce and described a prototype system for making online investment decisions with such a Web-based multi-agent DSS. In the pre-Web era, DSS was primarily used in an "island" mode. In e-commerce applications today, DSS can be an integral part of the digital environment and directly support the actions of the involved parties (Vahidov & Fazlollahi, 2003-2004).

CONCLUSION

Since their creation in the early 1960's, DSSs have evolved over the past 4 decades and continue to do so today. Although DSSs have grown substantially since its inception, improvements still need to be made. New technology has emerged and will continue to do so and, consequently, DSSs need to keep pace with it. Also, knowledge needs to play a bigger role in the form of decision making.

Shim (2002) emphasized that DSSs researchers and developers should (i) identify areas where tools are needed to transform uncertain and incomplete data, along with qualitative insights, into useful knowledge, (ii) be more prescriptive about effective decision making by using intelligent systems and methods, (iii) exploit advancing software tools to improve the productivity of working and decision making time, and (iv) assist and guide DSSs practitioners in improving their core knowledge of effective decision support.

The prior statement sums up the courses of action that need to be taken very well. The successful integration of DSSs and KMSs could revolutionize DSSs and propel it to even greater heights in the future. In closing, DSSs have a storied history that spans the course of 4 decades; however, the greatest mark may be made in the not so distant future as DSSs continue to evolve.

REFERENCES

Alter, S. (2004). A work system view of DSS in its fourth decade. *Decision Support Systems, 38*(3), 319-327.

Angus, J. (2003). *Does BA beat BI? New, predictive business analytics tools mean business can do much more than react to the vagaries of chance*. Retrieved August 25, 2007, from http://infoworld.com/article/03/08/29/34FEbusan_2.html

Arnott, D. (2004). Decision support systems evolution: framework, case study and research agenda. *European Journal of Information Systems, 13*(4), 247-259.

Arnott, D. (2006). Cognitive biases and decision support systems development: A design science approach. *Information Systems Journal, 16*(1), 55-79.

Arnott, D., & Pervan, G. (2005). A critical analysis of decision support systems research. *Journal of Information Technology, 20*(2), 67-87.

Ba, S., Stallaert, J., & Whinstone, A. B. (2001). Optimal investment in knowledge with a firm using a market mechanism. *Management Science, 47*, 1203-1220.

Banker, R. D., & Kauffman, R. J. (2004, March). The evolution of research on information systems: A fiftieth-year survey of the literature in management science. *Management Sciences, 50*(3), 281-298.

Bapna, R., Goes, P., & Gupta, A. (2003). Analysis and design of business-to-consumer online auctions. *Management Science, 49*, 85-101.

Benbasat, I., & Zmud, R. W. (1999). Empirical research in information systems: The question of relevance. *MIS Quarterly, 23*(1), 3-16.

Bolloju, N., Khalifa, M., & Turban, E. (2002). Integrating knowledge management into enterprise environments for the next generation decision support. *Decision Support Systems, 33*(2), 163-176.

Bui, T., & Lee, J. (1999). An agent-based framework for building decision support systems. *Decision Support Systems, 25*(3), 225-237.

Donciulescu, D. A., Filip, C. I., & Filip, F. G. (2002). Towards intelligent real-time decision support systems for industrial milieu. *Studies in Informatics and Control, 11*(4).

Eom, S. (1999). Decision support research: current state and trends. *Industrial Management & Data Systems, 99*(5), 213-221.

Geoffrion, A. (1987). An introduction to structured modeling. *Management Science, 33*(5), 547-588.

Goepp, V., Kiefer, F., & Geiskopf, F. (2006). Design of information system architectures using a key-problem framework. *Computer in Industry, 57*(2), 189-200.

Gnatovich, R. (2006). *BI versus BA. Analytics is a lot more than just a dashboard*. Retrieved August 25, 2007, from http://cio.co.nz/cio.nsf/UNID/0B520925A900C4BCCC25712D005DBF5E

Hess, T. J., Rees, L. P., & Rakes, T. R. (2000). Using autonomous software agents to create the next generation of decision support systems. *Decision Sciences, 31*(1), 1-31.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-106.

Keen, P. G. W., & Morton, M. S. S. (1978). *Decision support systems: an organizational perspective*. Reading, MA: Addison-Wesley Pub. Co.

Kelly, F., & Steinberg, A. (2000). A combinatorial auction with multiple winners for universal service. *Management Sciences, 46*(4), 586-597.

Klashner, S., & Sabet, S. (2007). A DSS design model for complex problems: Lessons from mission critical infrastructure. *Decision Support Systems, 43*, 990-1013.

Lauden, K. C., & Lauden, J. P. (2005). *Management information systems: managing the digital firm* (6th ed.). Upper Saddle River: Pearson Prentice Hall.

Lenard, M. L. (1993). An object-oriented approach to model management. *Decision Support Systems, 9*, 67-73.

Little, J. D. C. (1970). Models and managers: the concept of a decision calculus. *Management Science, 16*(8), 466-485.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM, 37*(7), 31-40.

March, S., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems, 15*, 251-266.

Nagarur, N. N., & Kaewplang, J. (1999). An object-oriented decision support system for maintenance management. *Journal of Quality in Maintenance Engineering, 5*(3), 248-258.

Nemati, H. R., Steiger, D. M., Iyer, L. S., & Herschel, R. T. (2002). Knowledge warehouse: An architectural integration of knowledge management, decision support, artificial intelligence and data warehousing. *Decision Support Systems, 33*(2), 143-161.

Orlikowski, W. J., & Iacono, C. S. (2001). Desperately seeking the "IT" in IT research—a call for theorizing the IT artifact. *Information Systems Research, 12*(2), 121-134.

Pillutla, S. N., & Nag, B. N. (1996). Object-oriented model construction in production scheduling decisions. *Decision Support Systems, 18*, 357-375.

Power, D. J. (2003, May 31). *A brief history of decision support systems*. Retrieved August 25, 2007, from http://DSSResources.com/history/dsshistory.html

Rowen, S. (2005). Decision support becomes a necessity. *Chain Store Age, 81*(2), 46.

SAS, Inc. (2004). *Software and services that give you the power to know. New business analytics report highlights SAS' leadership in scope*. Retrieved August 25, 2007, from http://www.sas.com/offices/europe/uk/press_office/press_releases/october2004/analytics.html

Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., & Carlson, C. (2002). Past, present, and future of decisions support technology. *Decision Support Systems, 33*(2), 111-126.

Sikder, I. U., & Gangopadhyay, A. (2002). Design and implementation of a web-based collaborative spatial decision support system: organizational and managerial implications. *Information Resources Management Journal, 15*(4), 33-47.

Srinivasan, A. & Sundaram, D. (2000). An object relational approach for the design of decision support systems. *European Journal of Operational Research, 127*(3), 594-610.

Stabell, C. B. (1983). A decision-oriented approach to building DSS. In J. L. Bennet (Ed.), *Building decision support systems*. Reading, MA: Addison-Wesley.

Thomsen, E. (2003). This year brought to you by the letter E. *Intelligent Enterprise*, 1, 3(1), 18-20.

Tian, Q., Ma, J., Liang, J., Kwok, R. C. W., & Liu, O. (2005). An organizational decision support system for effective R&D project selection. *Decision Support Systems, 39*, 403-413.

Vahidov, R., & Fazlollahi, R. (2003-2004, Winter). A multi-agent DSS for supporting e-commerce decisions. *The Journal of Computer Information Systems, 44*(2), 87-93.

West, L. A., Jr., & Hess, T. J. (2002). Metadata as a knowledge management tool: supporting intelligent agent and end user access to spatial data. *Decision Support Systems, 33*(2), 247-264.

KEY TERMS

Business Analytics (BA): A technological system that collects and evaluates all relevant data then scrutinizes it and puts it into different simulations to find out which one is the most appropriate.

Business Intelligence (BI): A system of technologies for collecting, reviewing, and hoarding data to assist in the decision making process.

Decision Support Systems (DSSs): An interactive, flexible, and adaptable computer-based information system, especially developed for supporting the solution of a non-structured management problem for improved decision making. It utilizes data, provides an easy-to-use interface, and allows for the decision maker's own insights

Interface (or User Interface): A component designed to allow the user to access internal component of a system, also known as the dialogue component of a DSS.

Knowledge Management: The distribution, access, and retrieval of unstructured information about human experiences between interdependent individuals or among members of a workgroup.

Sensitivity Analysis: Running a decision model several times with different inputs so a modeler can analyze the alternative results.

Software Agent: A software program that intelligently performs its duties without human inter-action.

Structured Modeling: A generic design strategy for representing complex objects that are encountered in modeling applications (Srinivasan & Sundaram, 2000).

Transaction Processing System (TPS): Computerized systems that perform and record the daily routine transactions necessary to conduct the business; they serve the organization's operational level.

Use Case: A complete sequence of related actions initiated by an actor; it represents a specific way to use the system.