

CHAPTER

# *One* Introduction

## *chapter* OVERVIEW

- 1.1 Introduction to Decision Support Systems
- 1.2 Defining a Decision Support System
- 1.3 Decision Support Systems Applications
- 1.4 Textbook Overview
- 1.5 Summary
- 1.6 Exercises

## 1.1 *Introduction to Decision Support Systems*

As a new graduate, equipped with the modeling and algorithmic skills taught in a standard operations management curriculum, Susan is ready to solve real-world problems. With a knowledge and understanding of theory and applications of mathematical programming, simulation techniques, inventory management, supply-chain management, and other industrial engineering (IE) or operations research (OR) and business topics, she is ready to help her company solve distribution problems by linear programming, inventory problems by applying the economic order quantity (EOQ) model, and manpower planning problems by integer programming. However, as she works on projects with her experienced colleagues and presents results to management, she realizes that she needs more than her models and equations. The management needs Susan to help solve decision problems, but they only want to know the final results of the analysis; they have no time or interest in understanding the mathematical model for these problems. They want this decision analysis tool to be available as a software system in which they can modify parameters and see different results for various scenarios. However, Susan is clueless about how to develop such a system. She knows the right model but she does not know how to package her model and how to present it with friendly graphical user interface. She feels that her education did not impart to her the skills she needs to meet her job requirements.

Susan is not the only one facing problems in her job as an operations research or business decision analyst. This is a widely prevalent problem which is not addressed in the current IE/OR or business curriculums. As OR practitioners and business analysts, students are support staff members and are required to build systems for non-OR users. They must know how to package OR/business models so that they can be comfortably used by top managers and other co-workers. Real-life decision making often requires building interactive systems, which students must know how to design and implement. To summarize, students must learn sufficient information technology skills so that they can build intelligent information systems, alternatively, called decision support systems, which can run sophisticated models at the back-end, but are friendly enough at the front end to be used comfortably by any user.

A decision support system (DSS) gives its users access to a variety of data sources, modeling techniques, and stored domain knowledge via an easy to use graphical user interface (GUI). For example, a DSS can use the data residing in spreadsheets or databases, prepare a mathematical model using this data, solve or analyze this model using problem-specific methodologies, and can assist the user in the decision-making process through a graphical user interface. Students are frequently being employed in positions that require developing DSS which are gaining widespread popularity. As more and more companies install enterprise resource planning (ERP) packages and invest in building data warehouses, those who are able to create decision technology driven applications that interface with these systems and analyze the data they provide will become increasingly valuable. Indeed, imparting DSS development skills, which combine OR/business skills with information technology (IT) skills, will make students highly sought after in the modern workplace.

Developing courses that teach our students how to build decision support systems has been a challenging task so far since it requires the availability of platforms which allow the integration of various technologies (data, models, codes, etc.). However, in the past few years, several platforms have become available which allow such integration. One such platform is Microsoft Excel. Excel, which is the most widely used spreadsheet package among managers and engineers, allows data storage and model building. Excel also has many built-in programs as well as many add-on programs available that allow optimization and simulation of various

models built in Excel. Excel also has a macro programming language, Visual Basic for Applications (VBA), which allows building GUIs and manipulating Excel objects. Thus, Excel provides a platform in which fairly sophisticated DSS applications can be built. This book imparts the skills needed to build such systems.

## 1.2 *Defining a Decision Support System*

A decision support system is a model-based or knowledge-based system intended to support managerial decision making in semi-structured or unstructured situations (Turban and Aronson, 2001). A DSS is not meant to replace a decision maker, but to extend his/her decision making capabilities. It uses data, provides a clear user interface, and can incorporate the decision maker's own insights. Some of the major DSS capabilities are the following:

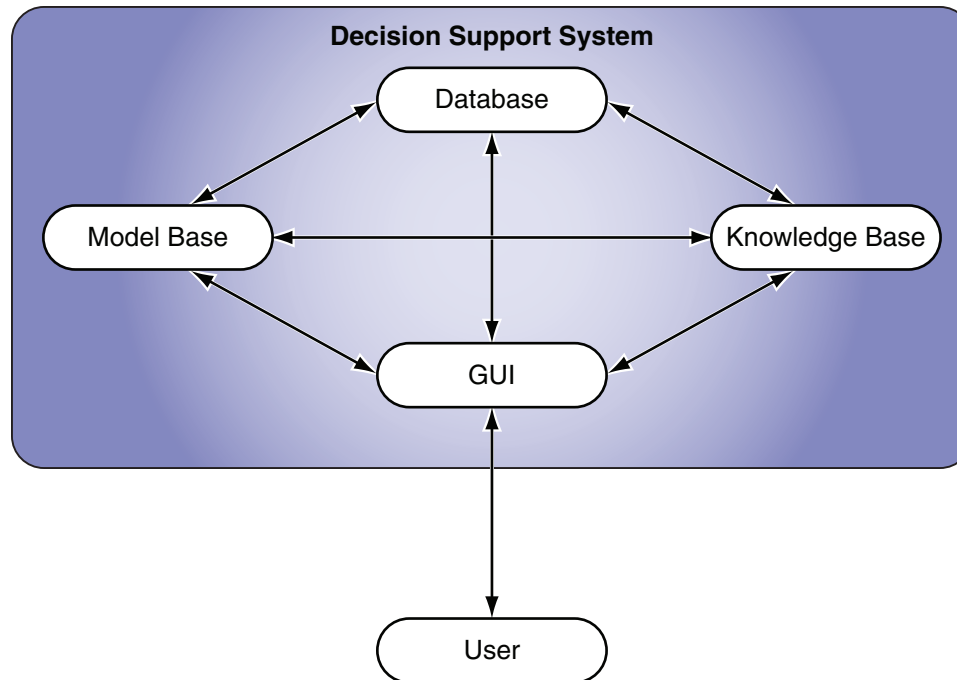
1. A DSS brings together human judgment and computerized information for semi-structured decision situations. Such problems cannot be conveniently solved by standard quantitative techniques or computerized systems.
2. A DSS is designed to be easy to use. User friendliness, graphical capabilities, and an interactive human-machine interface greatly increase the effectiveness of a DSS.
3. A DSS usually uses models for analyzing decision-making situations and may also include a knowledge component.
4. A DSS attempts to improve the effectiveness of decision making rather than its efficiency.
5. A DSS provides support for various managerial levels from line managers to top executives. It provides support to individuals as well as groups. It can be PC-based or web-based.

A DSS application contains five components: database, model base, knowledge base, GUI, and user (see Figure 1.1). The database stores the data, model and knowledge bases store the collections of models and knowledge, respectively, and the GUI allows the user to interact with the database, model base, and knowledge base. The database and knowledge base can be found in a basic information system. The knowledge base may contain simple search results for analyzing the data in the database. For example, the knowledge base may contain how many employees in a company database have worked at the company for over ten years. A decision support system is an intelligent information system because of the addition of the model base. The model base has the models used to perform optimization, simulation, or other algorithms for advanced calculations and analysis. These models allow the decision support system to not only supply information to the user but aid the user in making a decision. We now present a more detailed look at each of these components.

**Database:** The database provides the data with which decisions are made. The data may reside in spreadsheets or a data warehouse, a repository for relevant corporate decision-making data. The database allows a user to access, manipulate, and query data. Some examples of databases would include a spreadsheet containing personal banking account information or a data warehouse containing shipment records of various products.

**Model Base:** A model base contains statistical, financial, optimization, or simulation models that provide the analysis capabilities in a DSS. Some popular optimization models include linear

programming, integer programming, and nonlinear programming. The DSS allows the ability to invoke, run, and change any model or combine multiple models. An example of a model base would be an integer programming model used to solve a capital budgeting problem.



**Figure 1.1** A schematic view of a decision support system.

**Knowledge Base:** Many managerial decision making problems are so complex that they require special expertise for their solution. The knowledge base part of a DSS allows this expertise to be stored and accessed to enhance the operation of other DSS components. For example, credit card companies use a DSS to identify credit card thefts. They store in their knowledge base the spending patterns that usually follow credit card thefts; any abnormal activity in an account would trigger checking for the presence of those patterns and a possible suspension of the account.

**GUI:** The graphical user interface covers all aspects of communication between a user and a DSS application. The user interface interacts with the database, model base, and knowledge base. It allows the user to enter data or update data, run the chosen model, view the results of the model, and possibly rerun the application with different data and/or model combinations. The user interface is perhaps one of the most important components of a DSS because much of the flexibility and ease of use of a DSS are derived from this component.

**User:** The person who uses the DSS to support the decision making process is called the user, or decision maker. A DSS has two broad classes of users: managers and staff specialists, or engineers. When designing a DSS, it is important to know for which class of users the DSS is being designed. In general, managers expect a DSS to be more user-friendly than do staff specialists.

A DSS should be distinguished from more common management information systems (MIS). An MIS can be viewed as an information system that can generate standard and exception reports and summaries for managers, provide answers to queries, and help in monitoring the performance of a system using simple data processing. A DSS can be viewed as a more sophisticated MIS where we allow the use of models and knowledge bases to process the data and perform analysis.

## 1.3 *Decision Support Systems Applications*

Given the above definition of a decision support system, we have developed several spreadsheet-based DSS applications in Excel. Using the spreadsheet functionality in Excel and the VBA programming capabilities, a complete decision support system can be developed. As an IE/OR or business graduate, students will discover opportunities in their careers to develop and use DSS applications. Many students who have taken DSS courses have reported that their skills of combining modeling with an information technology packaging have truly helped them be outstanding at their jobs. Let us consider two examples of DSS applications which may be found in industry.

**Car Production:** Consider a factory which produces cars. The manager of the factory, probably a business or IE/OR graduate, may need to make some large-scale decisions about ordering parts, hiring/firing employees, finding new suppliers, or making changes to the production process. Let us focus on a production process DSS application. The manager may be deciding where to place a new piece of equipment or how to add a new product part to the production sequence. With the use of some basic simulation and analysis tools, one could develop a DSS which allows the manager to enter the parameters to describe a possible scenario and see how it would affect production. The manager may not want to know the details of the models used, but rather what would be the affect on cost and production time and quantity if a specific change was made. This is just one example of how a DSS could be used to aid in the car production industry.

**Railroad Car Management:** Consider a railroad company which owns several trains on which they place several thousand railroad cars which ship to several cities in the country. A distribution manager, again with an IE/OR or business background, may need to decide which cars should go on which trains to which cities. He would benefit from using an optimization model which allows him to modify certain constraints or focus on various objectives and compare the resulting distribution plans. He may want to display the car and train plans visually, may be projected on a country map, to have a better understanding of the effects of one solution compared to another. A DSS would aid him in accomplishing this analysis and making a decision which considers all scenarios and possible outcomes.

The applications we develop are basic illustrations of decisions which are made in IE/OR and business industries. Two examples, selected from the DSS applications developed in Part III of this book, are described below.

**Portfolio Management and Optimization:** In this application, we allow users to create and/or edit their portfolio as well as optimize their investments. To create a portfolio, users can choose from a list of stocks in a database in Excel and add or remove them to/from their portfolio. The stocks are listed by name and category. These stocks can also be compared using their

historical annual returns and changes in market price. Once users have created their portfolio, they may also edit it at any time. Users may then optimize their investment by specifying a desired return on their portfolio and the amount available for investing. The application then solves an optimization problem which minimizes users' risk on their selected portfolio. The suggested investment strategy is then displayed to the users. This application allows users to interact with the database of stock information and the knowledge base of comparative statistics. The model base for this DSS uses the Excel Solver to perform an optimization which minimizes the user's risk on the portfolio investments.

**Facility Layout:** In this application, we study a facility location problem which consists of placing  $n$  facilities at  $n$  locations to minimize the total handling cost. This problem is also known as the Quadratic Assignment Problem (QAP). The QAP arises in many other applications, such as the allocation of plants to candidate locations, the backboard wiring problem, design of control panels and typewriter keyboards, turbine balancing, etc. The user begins the application by providing the size of their facility. From the dimensions provided, a layout is displayed and a distance matrix is created. Random flow matrix values are generated which the user may overwrite if desired. From these two matrices, the cost matrix is derived; the total of the costs from this matrix is minimized by performing a pair-wise local search on the user's facility. The user may also opt to fix some facilities so that they cannot be moved when the local search is performed. The user may run this local search algorithm automatically or participate in the decision taken at each iteration. The final layout is then displayed to the user. The model base for this application uses an algorithm developed in the VBA code. This DSS application aids a facility designer in creating a facility layout which minimizes total handling cost. It can be used to solve typical facility layout problems which may be encountered by plant managers, school administrators, or in other applications. This DSS can also be used as a pedagogical tool in facility planning courses to illustrate the pairwise local search technique.

## 1.4 Textbook Overview

We now present an overview of the three parts of the text: Part I Excel Basics, Part II VBA for Excel, and Part III DSS Case Studies.

### 1.4.1 Overview of Excel

Microsoft Excel spreadsheets have become one of the most popular software packages in the business world, so much so that business schools have developed several popular Excel based courses. A spreadsheet application has functionality for storing and organizing data, performing various calculations, and using additional packages, called Add-Ins, for more advanced problem solving and analysis. *Part I* of this book discusses two aspects of Excel: basic functionality and extended functionality.

The topics we discuss on Excel basic functionality include referencing and names, functions and formulas, charts, and pivot tables. These are standard tools that may be common to most spreadsheet users. The topics we discuss on Excel extended functionality include statistical analysis, mathematical programming, simulation, and querying large data. We introduce Risk Solver Platform, an Excel add-in, to add in building simulation and optimization models. These tools are especially important for building a decision support system. The ability to model a

problem and solve it or simulate it adds the model base component of the DSS we are building. There are several examples given in the chapters in this part of the book which illustrate how these tools can be applied. It is important that the reader become familiar with the capabilities of Excel so that they know what they can offer the user when developing a decision support system.

### 1.4.2 Overview of VBA for Excel

VBA for Excel is a programming language that allows for further manipulation of the Excel functionalities. VBA for Excel also allows the developer to create dynamic applications which can receive user input for the model base component of the DSS. VBA allows users without knowledge of Excel to be able to use spreadsheet-based DSS applications. *Part II* of this book teaches important features of VBA for Excel.

We begin the VBA part of the book by illustrating the idea of macros and the visual basic environment. Then, we discuss how to work with variables, procedures, programming structures, and arrays in VBA. VBA for Excel is an easy to understand programming language. Even if the reader has not programmed before, they should be able to program several types of applications after reading these chapters. The student will learn how to program in VBA for Excel, which will enable them to quickly learn VBA for other Microsoft applications such as Access or Outlook as well as give them the basics for programming in any language. We then show the reader how to create a user interface in VBA. This discussion includes building user forms, working with several different form controls, using navigational functions, and designing a clear and professional application. VBA is beneficial as it places all of the complicated spreadsheet calculations and any other analysis in the background of a user-friendly system.

We then revisit some of the extended Excel functionality topics from Part I of the book. We show how VBA can enhance the modeling, simulation, and query features of Excel. Each of these chapters includes an application of a small DSS which combines the tools taught in VBA with the functionality taught in Excel. These techniques are especially important to understand in order to build complete DSS applications.

### 1.4.3 Overview of Case Studies

*Part III* of the book illustrates the relevance and importance of decision support systems in the fields of industrial and systems engineering, business, and some general engineering. We strive to accomplish this by showing how to develop DSS applications which integrate databases, models, methodologies, and user interfaces.

We have developed over 25 case studies. We have included some of these in the book and made the remaining case studies available at the website: [www.dssbooks.com](http://www.dssbooks.com). Most of the case studies consist of developing a complete decision support system and are based on an important application of IE/OR or business. We have also included some simpler case studies which apply to general engineering concepts. Through these case studies, students will learn how IE/OR and business techniques apply to real-life decision problems and how those techniques can be effectively used to build DSS applications.

Some of our case studies include portfolio management and optimization, facility location, queuing systems, critical path method (or project management), and a student information system. There are also case studies on forecasting, inventory scheduling, supply chain management, and capital budgeting. These case studies are just some of the numerous case studies we may develop in order to illustrate how DSS applications can be developed by combining



information technology tools with operations research and business tools to solve important decision problems. Extensions are listed for each case study for students to attempt or to use as ideas for other projects.

In this part of the book, we also propose a general DSS development process (Chapter 22). After the reader has learned how to work in the Excel spreadsheet environment (Part I) and is able to program in VBA to manipulate Excel objects and perform advanced calculations and analysis (Part II), they must then be able to combine these tools to develop a complete DSS application. Chapter 22 describes to the reader how to plan the DSS model and create an overview of the application, prepare the worksheets, create the user interface, and write the procedures so that the user's objectives are met and the decision maker will truly be aided by the DSS system they will use. In Chapter 23, we discuss how to design good graphical user interfaces and in Chapter 24, we discuss how to use good programming principles while developing a DSS. We give several examples of GUI designs to illustrate the importance of clarity and consistency when interacting with the user. The programming principles discussed will help the reader avoid errors in the DSS coding. These are both important chapters for the reader to understand before developing complete DSS applications.

#### 1.4.4 Overview of Appendices

We have three Appendices chapters at the end of the book. The first chapter, Appendix A, gives an overview of various Excel Add-In programs. We review the Analysis Toolpak, covered in Chapters 7 and 9, and the Standard Excel Solver. We also explain how to use the Premium Solver and present a comparison between the Premium and Standard Solvers. We then give an overview of some statistical and simulation Add-Ins, namely CPLEX, @RISK, Crystal Ball, and StatTools. The next chapter, Appendix B, describes how to perform debugging and error checking in VBA for Excel. We present several methods that can be used to prevent errors and check for errors. This may be a very useful appendix to refer to while learning the VBA topics discussed in Part II of the book. In the last chapter, Appendix C, we discuss advanced programming topics. One of these topics is object-oriented programming using class modules. We describe how class modules can be used to define new objects, properties, methods, and events. Another advanced programming topic is learning how to call outside applications from within a VBA procedure. This may be useful for running more complicated software or allowing the user an external interface to a needed feature not available in Excel.

## 1.5 Summary

- Decision support systems are model-based or knowledge-based systems which support managerial decision making; they are not meant to replace a decision maker, but to extend his/her decision making capabilities.
- There are five components to a DSS: data-base, model base, knowledge base, GUI, and user.
- Excel is a spreadsheet application with functionality for storing and organizing data, performing various calculations, and using additional packages for more advanced problem solving and analysis.
- VBA is a programming language that allows for further manipulation of the Excel functionalities and creation of dynamic applications which can receive user input for the model base component of the DSS.
- The case studies are intended to show the reader how to develop DSS applications which integrate databases, models, methodologies, and user interfaces.



## 1.6 Exercises

### 1.6.1 Review Questions

1. What are the components of a decision support system?
2. What is the difference between an information system and a decision support system?
3. What are some industrial engineering or business problems that may use a spreadsheet application like Excel to organize data? What kind of analysis might be done with this data using Excel functionality?
4. What user interface would be necessary to communicate with a user who does not have a background in Excel or VBA programming?
5. What are some applications of industrial engineering for which a decision support system could be built?

