Preface

Chapter OVERVIEW

- Motivation
- Book Contents
- Required Background
- Suggestions for Instructors
- Changes Made in the Second Edition
- Website Contents
- Acknowledgements

Motivation

Developing Spreadsheet-Based Decision Support Systems is intended to be a textbook which describes spreadsheet functionality and modeling in Microsoft Excel, illustrates the programming basic and advanced topics in Visual Basic Applications (VBA) for Excel, and applies these techniques to build decision support systems arising in several disciplines. This book can also be used as a self-learning reference book by practitioners.

Information systems based on decision models are called Decision Support Systems (DSS). A DSS uses the data residing in spreadsheets and/or databases, models it, processes or analyzes it using problem-specific methodologies, and assists the user in the decision-making process through a graphical user interface. Industrial Engineering and Operations Research (IE/OR) and Business School graduates are frequently being employed in positions that require developing DSS. We believe that decision support systems will gain widespread popularity in the future, and knowing how to develop such systems will make our students highly desirable in the workplace.

IE/OR departments and business schools teach their students decision model-based courses that impart spreadsheet modeling, optimization and simulation skills. Most of these departments require their students to take a computer programming course, often in C++, Java or Visual Basic. Several departments require their students to take a database course as well. Thus, students acquire background in modeling, optimization, simulation, database, and programming, but there are no courses that teach students how to integrate the technologies learned in these different courses to build complete decision support systems. Students are given many components, but they are not taught how to assemble them into a complete package suitable for usage by a customer. Decision support systems combine these technologies and are ideally suited to be an integral part of the IE/OR and business school curriculum.

Developing courses that teach our students how to build decision support systems has been a demanding task so far, since it requires the availability of platforms that allow the integration of various technologies (data, models, and programming). However, in the past decade, several platforms that allow such integration have become available. One such platform is the Microsoft Excel spreadsheet package. Spreadsheets allow data analysis, mathematical modeling, optimization, and simulation, and they have emerged as one of the most popular software packages engineers and managers use in their workplace. Spreadsheet modeling courses have become standard courses in many departments, and a need is now felt for the next level follow-up course. Microsoft Excel has a built-in programming language called *Visual Basic Applications* (VBA) for Excel. Excel combined with VBA for Excel provides a complete environment for building a spreadsheet-based DSS. VBA for Excel provides a mechanism for data analysis, implementing optimization and simulation models, and building graphical user interfaces. We have developed a complete set of course material in the Excel framework for a course on Spreadsheet-Based Decision Support Systems. This textbook is a byproduct of this effort. One can alternatively build decision support systems involving databases and using more sophisticated web-enabled technologies that are the subject of another parallel effort that led to the companion book *Developing Web-Enabled Decision Support Systems,* by Abhijit A. Pol and Ravindra K. Ahuja.

DSS development skills are important to all IE/OR and business school graduates for several reasons. First, having these skills will allow them to develop simple tools that might increase their productivity. Second, many graduates will be involved in information systems development during their career, and having a better understanding of the underlying technology will allow them to play a more useful role in the development process. Third, our graduates can develop a prototype system including the techniques learned here to establish a system proof-of-concept before it is turned over to professional programmers. Fourth, several graduates are often employed in departments that are service organizations catering to the modeling and optimization needs of other departments such as manufacturing, sales, marketing, accounting, and purchasing. These departments often request decision tools to be built to meet their business needs. The users in this department are not modeling experts, and the mathematical complexity of these models becomes an impediment to their use. If our graduates are well versed in the DSS building process, they can build optimization and simulation models, and package these models within friendly interfaces so that the modeling complexity is hidden from the user. This will make modeling and optimization approaches more popular in the workplace. Finally, with data mining becoming increasingly important, companies are realizing an important need for persons who understand data, models, algorithms, and information technology; our graduates can fulfill this critical need. Our students possess modeling and algorithmic skills to analyze the data but lack information systems development skills, and providing them with these skills will make them ideally suited for this task.

Book Contents

In order for the reader to learn how to develop a spreadsheet-based decision support system, we must teach them how to use Excel functionalities and the programming language-VBA for Excel. We also need to illustrate through some practical applications how to build such decision support systems. Currently, there are no books available in the market that include all three of these topics comprehensively. Our book is intended to meet this need. There are three parts of this textbook: learning Excel functionality, learning how to program with VBA for Excel, and learning how to develop DSS applications through several case studies.

Part I—Excel Essentials: This part presents an overview of Excel basic and extended functionalities. The basic functionality topics include referencing and names, functions and formulas, charts, pivot tables, and other Excel basics. The extended functionality topics include statistical analysis, Risk Solver Platform for Education for modeling and solving optimization and simulation problems, and working with large data. The purpose of this part of the book is to give the reader an idea of the tools available in Excel which span beyond simple spreadsheet functionality to include modeling, optimization, simulation, and advanced analysis. These tools are an important component of DSS application development.

Part II—VBA for Excel: This part presents an overview of programming in VBA and manipulating Excel objects. An introduction to the visual basic environment is given and macros are explained. The programming topics discussed include variables, procedures, programming structures, and arrays. User interface development is then discussed through the explanation of user forms, form controls, and navigational buttons. In the last few chapters of this part of the book, extended Excel functionality topics are revisited to illustrate the enhancement that VBA offers: optimization and simulation using Object Oriented API in Risk Solver Platform are revisited, and working with large data in VBA is revisited. The programming topics discussed are essential to the development of DSS applications.

Part III—Case Studies: This part presents several fully developed DSS applications arising in IE/OR, business, and general engineering. The case studies are preceded by three chapters which explain the DSS development process and provide more detailed instructions on designing a good user interface and using programming principles in VBA. It is important for the reader to understand how to plan the DSS application, prepare the spreadsheet, and implement the code so that the user's objectives are met and the decision maker will truly be aided by the DSS system they will use. We have developed 25 case studies covering a variety of DSS applications; however, due to space limitation we were not able to include all of the case studies in this printed form of the book. We have included 10 case studies in this book and the remaining case studies are available on the book website: www.dssbooks.com.

This book explains all topics through classical examples selected from IE/OR, business school, and engineering curriculum. Each new concept or idea is illustrated through examples and reinforced through exercises at the end of the chapter. We have also created an extensive list and description of possible student projects, which will further enhance students' learning experience.

Required Background

We assume that the reader has some basic Excel experience. We provide some links on the book website for online tutorials on Excel for those who do not have the required experience. The book is sufficiently comprehensive in its coverage on Excel functionalities that even experienced Excel users will find the material presented educative. We have, however, not tried to cover every Excel feature as it would have taken too much space. We have covered just enough topics in just enough details so that decent decision support systems can be built.

We expect some operations research (OR) and modeling experience in Chapter 8 of the book (solver and mathematical programming). A DSS course is typically offered to junior/senior level students, and an OR/modeling course should be a prerequisite, so most students should have that background. However, we do provide several examples in this chapter so that even a reader with minimal modeling experience should be able to understand how to formulate an optimization problem in the spreadsheet and use Risk Solver Platform to find the optimal solution.

The book does not require any programming experience. We instruct the reader on how to create variables, define functions, use basic programming structures, and work with arrays in VBA. The reader will also learn how to create good graphical user interfaces in Excel through user forms and various controls. It is useful if the reader has some programming experience, but it is not required. As with *Part I* of the book for experienced Excel users, some experienced programmers may skip some chapters in *Part II* of the book. However, it is good to review the programming examples in VBA. These chapters also include several applications which may be considered as “mini” DSS applications. Therefore, even for an experienced programmer, it is useful to review these chapters to strengthen the reader’s skills in using VBA in the context of developing a DSS.

We hope that after learning the Excel background and VBA material in *Parts I* and *II* of the book, the reader will be able to develop any of the case studies found in *Part III*. The case studies construct prototype decision support systems that are simple enough to be easily understood by the reader and complex enough to be reasonably accurate representatives of real-world problems. The case studies require that the reader has learned good VBA programming skills from Part II of the book.

Suggestions for Instructors

This book is primarily intended as a textbook for undergraduate and graduate students in the IE/OR and business school curriculums. This book can also be used as a self-study manual. DSS are great tools for consulting, and consultants can use the skills learnt profitably in developing their consulting practice.

As a textbook, this book can be used in a variety of ways to teach different courses. For an undergraduate-level course, instructors can cover topics at slower pace. In a graduate-level course, instructors can spend less time on Excel functionalities and cover more case studies. Instructors can offer semester-long courses (covering all the material), or half-semester/quarter courses on VBA for Excel and case studies. We present a variety of case studies from simple to complex, and the instructor can select the case studies that best suit the time frame and background of their students.

The material can also be used to supplement the courses currently taught. For example, several IE/OR departments and business schools offer spreadsheet-based Operations Research courses. The instructor can cover a module on VBA for Excel within the course and show how a simple Excel application can be easily turned into a powerful DSS. Similarly, logistics and supply-chain courses are becoming standard courses in the curriculum. The instructor can illustrate in a few class hours how to build a decision support system based on some decision problem in inventory, distribution, or transportation, and then assign student projects on building similar applications. Thus, this textbook can be used in a variety of ways, from teaching new courses to supplementing existing courses.

The spreadsheet-based DSS course may be taught in different formats including or excluding several different chapters from our text. We propose that the general structure of the course should begin by teaching Excel functionality to students to ensure they are familiar with the spreadsheet environment; then teach VBA programming to show students how to work with variables and programming structures as well as how to create a user interface; the course should then end with a full discussion of decision support systems and instructing students how to combine their acquired Excel and VBA skills to develop a DSS application. The text has been designed to follow this general course structure.

From our experience, it seems most productive to hold this course in a computer laboratory or require students to bring laptops to the class. We recommend the instructor to illustrate concepts with hands-on examples on the computer screen while students are watching and trying to do it themselves on their computers. We also recommend that there be a teaching assistant available to help students as they are doing the hands-on examples on their personal computers while the instructor illustrates the examples simultaneously at the front of the class. We have found that students learn much more in this manner as they experiment with Excel and VBA themselves along with the instructor and teaching assistant readily available to answer questions or address their difficulties.

In this suggested setting of a computer laboratory or class with laptops, we also suggest that the course be taught in two-hour sessions instead of one-hour sessions. This allows students enough time to set up their computers and instructors enough time to illustrate a full example during the class. The course material can be adjusted to teach a semester course, a half-semester module or a quarter course. The material can be covered in different rigors and at different paces and some of the material can be assigned for self-study.

We have discovered by teaching these courses over the years that students learn the most by doing course projects. Lectures teach them the technology and how to use it, but unless they apply it themselves to build complete systems, they do not assimilate the material. In addition, the process of developing a full system from conception to completion and seeing the fruits of their labor gives them tremendous satisfaction and confidence. Course projects may be done by teams of students, in which case they promote teamwork—an essential skill in any workplace. We have developed over 100 course projects from different application areas in IE/OR, business as well as engineering curriculums. These projects are available on the book website. Course projects can be assigned on an individual basis or in groups depending on the course size and course format. Students can select a project from this list or they can create their own project as long as it is sufficiently interesting and challenging. We require our students to present these projects before the entire class when completed. Many students have told us that doing these projects and building complete decision support systems was the most educative and learning experience for them in the course.

Changes Made in the Second Edition

We made two major changes to this new edition of the book. First, we feature Microsoft Excel 2010. Therefore, we have updated all of the Excel features and examples to reflect the changes made to Excel's interface and functionalities. Second, we introduce Risk Solver Platform for Education to model and solve optimization and simulation problems.

Microsoft Excel 2010 is more powerful and offers a number of additional functionalities as compared to its predecessors. This new version of Excel supports the processing of larger data sets. Excel spreadsheets now support data sets with up to 1.1 million rows and 16,000 columns. This is a large improvement compared to Excel 2003 which supported spreadsheets with up to 65,536 rows and 256 columns of data.

However, the most drastic change in Excel 2007 and 2010 as compared with prior versions of this product is the user interface. With Office 2007, Microsoft introduced an entirely new interface. Menus and toolbars are replaced with the Ribbon. Excel 2010 introduces one additional tab on the Ribbon, the File menu. Most Ribbon tabs contain commands that you use when working with your file. The File menu includes commands you use when you are finished working with the file, such as, save, open, recently used files, etc. In Chapter 2 we give an overview of the Ribbon and Quick Access Toolbar to help readers become familiar with these features and understand their functionalities. Based on our experience, the Ribbon is a user-friendly interface with many functional benefits.

Excel 2010 also provides more and better formatting options. In Chapter 2 we discuss conditional formatting and give examples of how to use new conditional formatting features such as icon sets, color scales, and in-cell bar charts. In this chapter we also discuss how to use different collections of Excel's new built-in themes to format a workbook. In Chapter 3 we discuss the new and improved charting interface of Excel and introduce Sparklines. We give a number of examples to highlight the benefits of using sparklines.

Excel 2010 also has improved capabilities for working with pivot tables. In Chapter 6 we give examples to introduce Slicers which are visual graphical filters you can use to filter the data in a pivot table. The new Excel commands now make it easier to create and modify a pivot table.

Excel 2010 also has an improved function library. The number of built-in functions available in Excel has increased to 400 as compared to 255 offered by Excel 2003. This increase is mainly due to the inclusion in Excel 2010 of about 89 functions available only to users that had access to Analysis ToolPak in earlier versions of Excel. In Chapter 4 we give an example of how to use the new IFERROR function of Excel. Excel has also improved the accuracy of some of its statistical distribution, financial, and math functions. The consistency of the function names has also been improved. For example, many of the distribution functions now include a dot in their name (NORM.DIST, NORM.S.DIST, NORM.INV, and NORM.S.INV) to distinguish between distribution and inverse distribution functions. Similarly, the change in name of statistical functions VAR, VARP, STDEV and STDEVP to VAR.S, VAR.P, STDEV.S and STDEV.P make it easier for the reader to understand their functionality. These function updates are reflected in Chapters 4 and 7.

In Chapter 10, we introduce the Official Excel Table. The new *Table* tools of Excel make the task of organizing and manipulating data in tables much easier. We give an example of how to use the tool when working with large data sets. We discuss the new *Remove Duplicates* command which removes any duplicates that may result when we combine together different sets of data.

In this new edition of the book we introduce Risk Solver Platform for Education, a trademark of Frontline Systems Inc. This is a new add-in for Excel that provides a variety of tools one can use to perform optimization, simulation, and sensitivity analysis, as well as build decision trees. Our motivation for using this product is twofold. First, Risk Solver Platform improves optimization and simulation capabilities of Excel. We can now solve larger problems faster (see Appendix A for a comparison of this product with the Standard Solver of Excel). Risk Solver Platform offers algorithms to solve linear, mixed integer, and smooth and non-smooth nonlinear programming problems. Second, the interface of Risk Solver Platform is user friendly and provides a variety of commands one can use to build a problem, solve the problem, and analyze the results.

In Chapters 8 and 9 we show how to use Risk Solver Platform to model and solve spreadsheet-based optimization and simulation problems. In Chapters 19 and 20 we discuss how to use Object Oriented API with Risk Solver Platform to modify and solve optimization and simulation models using VBA commands. We have also updated most of the case studies in Part III of the book to use Risk Solver Platform instead of the Standard Solver for solving optimization problems. The use of Risk Solver Platform has especially improved the coding efficiency and performance of the case studies that require simulation modeling and analysis; such as, Single and Multiple Server Queuing, Retirement Planning, Birthday Simulation, and Reliability Analysis. Instead of "manually" generating random problem inputs within For, Next loops in VBA to simulate a system, we now use Object Oriented API to read an existing simulation spreadsheet model, modify problem inputs, and collect a variety of statistics.

Website Contents

We have developed a website for this textbook which contains valuable resources for both students and instructors. The URL of this website is:

www.dssbooks.com

This website contains the following material:

- Excel files for the examples covered in all chapters
- Excel files for all the Hands-On Exercises covered in all chapters
- Excel files for all the 25 case studies developed by us
- Additional chapters describing the 15 case studies which were not included in the book
- PowerPoint presentations for all book chapters on Excel and VBA for Excel
- License for the educational version of Risk Solver Platform
- A booklet containing about 100 student projects
- Sample course schedules

The Solutions Manual for the book exercises is also available and will be provided to instructors offering courses using this book as the principle textbook. The website provides the email addresses for requesting the Solution Manual and giving your feedback to the book authors.

Acknowledgements

There are many people whom we would like to thank for making significant contributions to this book writing project. First and foremost, we would like to thank Dr. Donald Hearn, Chair of the Industrial and Systems Engineering Department at the University of Florida, who has been the driving force in this book-writing initiative. He motivated us to teach courses that incorporate greater levels of information technology in the IE/OR curriculum at the University of Florida. He also inspired us to write this book and provided constant encouragement throughout its evolution from concept to reality. We are truly thankful to him for his encouragement and support.

Several students helped us in the development of the book at different stages. These students included both undergraduate and graduate students of the Industrial and Systems Engineering and English Departments at the University of Florida and Mississippi State University. In particular, we would like to thank Richard Barrow, Melissa Sullivan, William Bowen, Preston Cauley and Gokçe Palak for the contributions they made in developing exercises and case studies; Carolyn Houston, Krystal Harriot, Sarah Schiff, Michael Smith, and Tamara Johnson for their copy editing of the text; and Ali Abbas, Krishna Jha, and Burak Ekşioglu for developing team projects. Our special thanks go to Ashish Nemani who assisted us in preparing the index and proofreading the book. We would also like to thank Guvenc Şahin and Onun Şeref for their feedback from teaching a DSS course using the material we developed. We are also appreciative of the feedback provided by students enrolled in these initial DSS courses. Several students, who assisted us in the book-writing project, were supported by the National Science Foundation Course Curriculum Development Grant 0341203.

Next, we would like to thank the Frontline Systems team for continuously supporting us with advice when revising several chapters of this new edition. In particular, we would like to thank Edwin Straver for never failing to respond very quickly to our questions and graciously offering his expert opinion.

Finally, we thank our families for their constant support and encouragement.

Sandra D. Ekşioğlu
Michelle M.H. Şeref
Ravindra K. Ahuja
Wayne L. Winston