

Preface

OVERVIEW

- Motivation
- Book Contents
- Required Background
- Suggestions for Instructors
- Web site Contents
- Acknowledgments

Motivation

Developing Web-Based Decision Support Systems is intended to be a textbook that describes database design and development, illustrates the programming topics (basic as well as advanced) in Visual Basic .NET and database connectivity, demonstrates Web-enabling using ASP .NET, and applies these techniques to building decision support systems arising in several disciplines. This book can also be used as a supplementary reference book for courses or as a self-study manual by practitioners.

Information systems based on decision models are called *Decision Support Systems* (DSS). A DSS uses the data residing in 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 its students decision models-based courses that impart spreadsheet modeling, optimization, and simulation skills. Most of these departments require its students to take a computer-programming course, often in C++, Java, or Visual Basic (VB). Several departments require its 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 cannot assemble them into a complete package suitable for use by a customer. Decision support systems combine these technologies, and teaching how to build such systems is 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 Microsoft's .NET technology.

The *Visual Basic .NET* (VB .NET) package of the .NET platform can be used in conjunction with *Microsoft Access* databases to build Windows-based, single-user database applications. The Microsoft Access database allows data storage, and VB .NET—an easy-to-learn yet powerful object-oriented programming (OOP) language—provides a mechanism for data analysis, implementing optimization and simulation models, and building graphical user interfaces. Further, Web programming has also become easier with the introduction of *Active Server Pages .NET* (ASP .NET). This .NET package makes use of standard *HyperText Markup Language* or HTML (the front-end) and VB .NET (the back-end) to deploy elegant and easy-to-build database-connected, Web-enabled applications. Another powerful .NET package is *Crystal Report .NET*. The Crystal Report package provides an ideal environment for creating reports and charts that can effectively summarize data. Thus, we have developed a complete set of course material in the Access and .NET framework for a course on Web-Enabled Decision Support Systems. This textbook is a byproduct of this effort. One can alternatively build spreadsheet-based decision support systems using the Microsoft Excel spreadsheet package, which is the subject of our parallel effort and which led to the companion book, “*Developing Spreadsheet-Based Decision Support Systems*,” by Michelle M.H. Seref, Ravindra K. Ahuja, and Wayne L. Winston.

The 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 using the techniques learned here to establish the system proof-of-concept before it is turned over to the 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, industries are realizing a critical need for persons who understand data, models, algorithms, and IT, and our graduates can fulfill this need. Our students possess modeling and algorithmic skills to analyze the data but lack information systems development skills, and providing them these skills will make them ideally suited for this task.

Book Contents

Building a Web-enabled, data-driven decision support system requires four skills: knowledge of database design, database development, database programming language, and Web programming. 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 covers all of the five areas. Our book is intended to meet this need. There are five parts of this textbook: learning database design; getting familiar with database development with Microsoft Access; learning how to program with VB .NET and build Windows-based database applications; making them Web-enabled with ASP .NET; and learning how to develop DSS applications through several case studies.

Part I—Principles of Good Database Design

We start this part of the book with an introduction to decision support systems and an introduction to databases (Chapters 1 and 2). We then cover principles of good database design. The database design process primarily involves two kinds of data modeling: Object-Based Modeling and Record-Based Modeling. The *entity-relationship (E-R) model*—an object-based data model—describes the data in the form of an E-R diagram, entities, attributes, and relationships and is the subject of Chapter 3. The *relational data model*—a record-based data model—is based on the mathematical concept of *relations* and uses tables (or relations) to represent data and relationships and is the subject of Chapter 4.

Part II—Database Development with MS Access

This part of the book introduces the database development process using the Microsoft Access package. In Chapter 5, we provide a tour of Access's development environment. We introduce the development window, various menus and toolbars, and the architecture of Access and its main components. In Chapter 6, we give a detailed explanation of Access tables, the central element of relational databases. Access tables store data and provide an effective grid structure for data organization. We also discuss how to design and modify Access tables in this chapter. Chapter 7 is dedicated to relationships among Access tables. Once we have populated them with data, we can query tables to retrieve useful information. We do this through Access queries, which are the subject of Chapter 8. Chapter 9 introduces SQL, the standard language used to write database queries. We have also written two comprehensive chapters on Access Forms and Access Reports, which are available on the book Web site: www.dssbooks.com.

Part III—Developing Windows Applications with VB .NET

With potential uses of a database application in mind, we present VB .NET as a database programming language in this part of the book. We first introduce, in Chapter 10, the Visual Studio environment—Integrated Development Environment (IDE)—that provides all the tools we

need to develop a database application. In Chapter 11, we introduce VB .NET along with its programming language constructs: variables, data types, control structures, operators, and arrays. Chapter 12 presents a discussion of the object-oriented support features of VB .NET, namely, Classes, Modules, Procedures, Methods, and Properties. In Chapter 13, we formally introduce Windows forms and controls. Forms and controls constitute the building blocks of a GUI. In Chapter 14, we discuss database connectivity using ADO .NET. In Chapter 15, we illustrate more advanced database connectivity topics. Finally, in Chapter 16, we introduce Crystal Reports, a powerful tool that can summarize data in the form of reports and graphs, and discuss its integration with Windows forms.

Part IV—Web Application Development with ASP .NET

The Internet era has taken information-sharing to new heights, allowing billions of users to share information on the World Wide Web (WWW). This development has created the need for Web-enabled applications that are accessible to a large number of users over a network. Today's information systems involving databases are pervasive, and a growing number of them are Web-enabled. In this part of the book, we illustrate the process of developing Web applications. We discuss Hyper Text Markup Language (HTML), a Web interface standard, in Chapter 17. We introduce a Web programming language, ASP .NET, in Chapter 18. We then introduce database connectivity in Web applications using VB .NET and ADO .NET in Chapter 19 to build a complete Web-enabled DSS.

Part V—Case Studies

This part presents five fully developed DSS applications arising in IE/OR, business, and general engineering. These case studies walk through the complete process of developing a Web-enabled decision support system. Through these case studies, students will learn how IE/OR and business techniques apply to real-life decision problems and how we can effectively use those techniques to build DSS applications. Our case studies include Online Book Store, Portfolio Management and Optimization, Television Advertisement Allocation, Voyage Prophecy, and ADO .NET and MS Excel. Each case study also lists extensions that students can do as additional projects or practice assignments. We have developed a few more case studies that are available on the book Web site: www.dssbooks.com. The space limitation did not permit us to include these case studies in this printed form of the book.

This book explains all topics through classical examples selected from IE/OR, business school, data management, and engineering curriculums. Each new concept or idea is illustrated through examples or hands-on tutorials and reinforced through exercises at the end of the chapter. Each chapter contains an in-class assignment, several review questions, and many hands-on exercises. We also have created an extensive list and description of student course projects (over 50 such projects), which will further enhance students' learning experience. These projects are available at the book Web site.

Required Background

The book does not require any prior experience with databases and can be used as an introductory text for novices in database design, development, and implementation. For readers familiar with the design process and the Access environment, this book provides a comprehensive review of these topics.

Prior experience in programming is not mandatory to use this book. We instruct readers on how to create variables, define functions, use basic programming structures, and work with arrays using Visual Basic language. Readers will also learn how to create good graphical user interfaces in the Visual Studio environment. This material also includes several applications that may be considered “mini” DSS applications. Therefore, even for an experienced programmer, it is useful to review these chapters to strengthen the reader’s skills in using VB .NET and ASP .NET in the context of developing a DSS.

We hope that after learning the database background and application development material in *Parts I–IV* of the book, readers will be able to develop any of the case studies found in *Part V*. The case studies construct prototypes of data-driven, Web-enabled decision support systems that are simple enough to be understood by undergraduates and complex enough to be reasonably accurate representatives of real-world problems. The case studies require that you have picked up good database design, development, and programming skills from *Parts I–IV* 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 reference book to supplement other textbooks in courses and as a self-study manual. DSS are great tools for consulting, and consultants can use the skills learned profitably in developing their consulting practice.

As a textbook, this book can be used in a variety of ways to teach different courses. It can be used to offer courses for undergraduate and graduate students. In an undergraduate-level course, instructors can cover topics at a slower pace. In a graduate-level course, instructors can spend less time on some topics such as programming basics or HTML, and focus more on case studies. Instructors can offer semester-long courses (covering all the material), or half-semester/quarter courses on either database design and development or Windows and Web application development.

The Web-enabled 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 begin by teaching database design and development to students to ensure that they are familiar with the databases; then teach VB .NET programming, database connectivity, and ASP .NET to show students how to work with Windows and Web-enabled database applications; the course should then end with a full discussion of decision support systems and instructing students how to combine their acquired database and programming 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 that the instructor illustrate concepts, hands-on examples, or tutorials on the computer screen while students are watching them and trying to do them themselves on their computers. We suggest that the instructor end each class by conducting one of the hands-on examples or tutorials as an illustration of all the concepts taught in that class. 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 may illustrate the example simultaneously at the front of the class. We have found that students learn much more in this manner as they experiment with Access, VB .NET, and ASP .NET themselves along with the instructor, with the 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 semester long courses or half-semester/quarter courses. 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 50 course projects from different application areas in IE/OR, business, and engineering curriculums. These projects are available at the book Web site. 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 our 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 experience for them in the course. We can't recommend them more highly!

Web site Contents

We have developed a Web site for this textbook that contains valuable resources for both students and instructors. The URL of this Web site is: www.dssbooks.com

This Web site contains the following material:

- Database files, VB .NET and ASP .NET source code, and application files for all the Hands-On tutorials covered in all chapters
- PowerPoint presentations for all book chapters
- Source code and the complete application package including database files for all five case studies discussed in Part V of the book
- Additional case study chapters that could not be included in the book
- Database files for Hands-On exercises for required chapters
- Additional chapters on Access Forms and Reports that could not be included in the book
- A booklet containing about 50 student projects
- Sample course schedules

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

Acknowledgments

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 a concept to a reality. We are truly thankful to him for his encouragement and support.

Next, we would like to offer many thanks to Sandra Duni Eksioglu. Sandra received her doctorate in Industrial and Systems Engineering from the University of Florida and has been working with us since the inception of the project. She has developed several exercises, their solutions, and team projects for the book. Sandra has taken a lead role in organizing all the team projects into a booklet and in preparing the solutions manual for the book. Her hard work and dedication is much appreciated.

Several students of the Industrial and Systems Engineering, Computer Science, and English Departments at the University of Florida helped us in the development of the book at different stages. In particular, we would like to thank Margaret Reece and Padmavati Sridhar for the contributions they made in developing several exercises and their solutions for the book; and Krishna Jha, and Burak Eksioglu for developing several team projects. Our special thanks to the programming junta of Rakesh Desai, Siddharth Gaitonde, Kunal Gandhi, Tapasvi Moturu, and Mohammad Jaradat for the contributions they made in developing case studies and examples for the book. Ruben Galbraith assisted us in preparing the PowerPoint presentations for the book chapters, and we appreciate his help. We extend our sincere thanks to Dean Swinford, Tamara Johnston, Carolyn Houston, and Krystal Harriot for their copy editing of the text. We would also like to thank students of the DSS courses offered at the University of Florida who served as guinea pigs to test early versions of book chapters and who gave valuable feedback. We are indeed indebted for their feedback. Several students who assisted us in the book-writing project were supported by the National Science Foundation Course Curriculum Development Grant 0341203.

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

Abhijit A. Pol
Ravindra K. Ahuja

