SYSTEMS ANALYSIS AND DESIGN

An Organizational Approach

nd McLeod, Jr.

SYSTEMS ANALYSIS AND DESIGN

An Organizational Approach

Raymond McLeod, Jr.
Texas A & M University

The Dryden Press
Harcourt Brace College Publishers
Fort Worth Philadelphia San Diego New York Orlando Austin San Antonio
Toronto Montreal London Sydney Tokyo

Acquisitions Editor	Richard J. Bonacci
Developmental Editor	Stacey Fry
Project Editor	Michele Tomiak
Art Directors	Beverly Baker, Sue Hart
Production Manager	Marilyn Williams
Photo & Permissions Editors	Steven Lunetta, Julia Stewart
Publisher	Elizabeth Widdicombe
Director of Editing, Design, and Production	Diane Southworth
Copy Editors	Carolyn Sweeney, Larry Bromley
Compositor	Clarinda Company
Text Type	Palatino
Cover Image	© 1993 Tim Brown, Tony Stone Worldwide

Address for Editorial Correspondence
The Dryden Press, 301 Commerce Street, Suite 3700, Fort Worth, TX 76102

Address for Orders
The Dryden Press, 6277 Sea Harbor Drive, Orlando, FL 32887
1-800-782-4479, or 1-800-433-0001 (in Florida)

ISBN: 0-03-055154-4

Permissions and other credits appear following the Index.

Printed in the United States of America

3 4 5 6 7 8 9 0 1 2 9 8 7 6 5 4 3 2 1

The Dryden Press
Harcourt Brace College Publishers

To Martha

The Dryden Press Series in Information Systems

Arthur Andersen & Co./Flaatten, McCubbrey, O'Riordan, and Burgess
Foundations of Business Systems
Second Edition

Arthur Andersen & Co./Boynton and Shank
Foundations of Business Systems: Projects and Cases

Anderson
Structured Programming Using Turbo Pascal: A Brief Introduction
Second Edition

Brown and McKeown
Structured Programming with Microsoft BASIC

Coburn
Beginning Structured COBOL

Coburn
Advanced Structured COBOL

Dean and Effinger
Common-Sense BASIC: Structured Programming with Microsoft QuickBASIC

Federico
WordPerfect 5.1 Primer

Goldstein Software, Inc.
Joe Spreadsheet, Macintosh Version

Goldstein Software, Inc.
Joe Spreadsheet, Statistical

Gray, King, McLean, and Watson
Management of Information Systems
Second Edition

Harrington
Database Management for Microcomputers: Design and Implementation
Second Edition

Janossy
COBOL: An Introduction to Software Engineering

Laudon and Laudon
Business Information Systems: A Problem-Solving Approach
Second Edition

Laudon, Laudon, and Weill
The Integrated Solution

Lawlor
Computer Information Systems
Third Edition

Liebowitz
The Dynamics of Decision Support Systems and Expert Systems

McKeown
Living with Computers
Fourth Edition

McKeown
Living with Computers with BASIC
Fourth Edition

McKeown
Working with Computers
Second Edition

McKeown
Working with Computers with Software Tutorials
Second Edition

McKeown and Badarinathi
Applications Software Tutorials: A Computer Lab Manual Using WordPerfect 5.1, Lotus 1-2-3, dBASE III PLUS and dBASE IV

McKeown and Leitch
Management Information Systems: Managing with Computers

McLeod
Systems Analysis and Design: An Organizational Approach

Martin
QBASIC: A Short Course in Structured Programming

Martin and Burstein
Computer Systems Fundamentals

Mason
Using IBM Microcomputers in Business: Decision Making with Lotus 1-2-3 and dBASE III PLUS

Millspaugh
Business Programming in C for DOS-Based Systems

O'Brien
The Nature of Computers

O'Brien
The Nature of Computers with Productivity Software Guides

Parker
Computers and Their Applications
Third Edition

Parker
Computers and Their Applications with Productivity Software Guide
Third Edition

Parker
Productivity Software Guide
Fourth Edition

Parker
Understanding Computers and Information Processing: Today and Tomorrow
Fifth Edition

Parker
Understanding Computers and Information Processing: Today and Tomorrow with BASIC
Fifth Edition

Robertson and Robertson
Microcomputer Applications and Programming: A Complete Computer Course with DOS, WordPerfect 5.1, Lotus 1-2-3, dBASE III PLUS (or dBASE IV) and BASIC

Robertson and Robertson
Using Microcomputer Applications (A Series of Computer Lab Manuals)

Roche
Telecommunications and Business Strategy

Simpson and Tesch
Introductory COBOL: A Transaction-Oriented Approach

Sullivan
The New Computer User

Swafford and Haff
dBASE III PLUS

Lab Manual Series from Electronic Learning Facilitators, Inc.
The DOS Book
The Lotus 1-2-3 Book
Stepping Through Excel 4.0 for Windows
Stepping Through PageMaker 5.0 for Windows
Stepping Through Windows 3.1
Stepping Through Word 2.0 for Windows
Up and Running with Harvard Graphics 1.03 for Windows
Up and Running with PageMaker 5.0 for Windows
Up and Running with WordPerfect 5.2 for Windows
Up and Running with Quattro Pro 1.0 for Windows
Up and Running with Microsoft Works 2.0 for Windows
Up and Running with Lotus 1-2-3 Release 1.1 for Windows
Up and Running with Paradox 1.0 for Windows
Up and Running with DOS 6.0
Up and Running with Paradox 4.0 for DOS
Up and Running with Microsoft Works 3.0 for DOS
Up and Running with Excel 4.0 for the Macintosh
Up and Running with Word 5.1 for the Macintosh
Up and Running with PageMaker 5.0 for the Macintosh
Up and Running with Microsoft Works 3.0 for the Macintosh
Working Smarter with DOS 5.0
Working with WordPerfect 5.0
Working with WordPerfect 5.1

Martin and Parker
Mastering Today's Software Series
Texts available in any combination of the following:
Microcomputer Concepts
Extended Microcomputer Concepts
Disk Operating System 5.0
Disk Operating System 6.0
Windows 3.1
WordPerfect 5.1
WordPerfect 5.2 for Windows
WordPerfect 6.0
Lotus 1-2-3 (2.2/2.3)
Lotus 1-2-3 (2.4)
Quattro Pro 4.0
dBASE III PLUS
dBASE IV (1.5/2.0)
Paradox 4.0
BASIC

Martin, Series Editor
Productivity Software Guide Lab Manual Series
Disk Operating System (DOS)
Windows 3.1
Word Processing with WordPerfect 5.1
Word Processing with WordPerfect for 5.2 Windows
Spreadsheets with Lotus 1-2-3
Spreadsheets with Quattro Pro 4.0
Database Management with dBASE III PLUS
Database Management with dBASE IV
Database Management with Paradox 4.0
A Beginner's Guide to BASIC

Harcourt Brace College Outlines

Kreitzberg
Introduction to BASIC

Kreitzberg
Introduction to Fortran

Pierson
Introduction to Business Information Systems

Veklerov and Pekelny
Computer Language C

PREFACE

One or more courses in systems analysis and design have always been required of both undergraduate and graduate students majoring in information systems. In these courses the students learn the tools and techniques that they will use when they become systems analysts.

Like everything else in the computer field, the processes of analyzing and designing systems are constantly changing. Although the systems theory, which underlies every aspect of systems work, has remained relatively constant, the technologies and methodologies are much different today than just a few years ago.

Although it would be possible to incorporate many of the current trends in a textbook on systems analysis and design, three appear to stand out as absolute requirements. They are the movement to end-user computing, increasing organizational influence, and use of computer-aided software engineering (CASE).

The Need to Recognize End-User Computing

The one change in the computer field that has the potential for affecting systems analysis and design more than any other is end-user computing (EUC), the situation in which users perform some or all of the work in developing their own systems. One of the many impacts of EUC has been a shift in the systems analyst's workload. The analyst is being relieved of many of the simpler applications and is assuming responsibility for a higher concentration of more advanced applications as well as a greater involvement in consulting activity.

College courses and textbooks in systems analysis and design must accommodate end-user computing, not only recognizing that it exists, but also preparing future information specialists to function in such a setting.

The Need for an Organizational Focus

The modern systems analysis and design course should also recognize another reality. This is the fact that systems development is no longer an activity aimed at meeting needs of *individual* users, or even the needs of *organizational subunits,* without considering the needs of the entire organization. Today's systems must be designed to meet the needs of the *organization.*

This organizational influence is reflected in the currently high level of interest in topics such as competitive advantage, strategic planning for information resources, information resources management, ethical computer use, and security of the information resources investment. Systems analysts must be aware of this organizational influence and be able to work within it.

The Need to Incorporate State-of-the-Art CASE Tools

During the forty-odd-year history of business systems, the process of development has evolved through four stages of refinement—traditional, structured, toolkit, and CASE.

The Traditional Approach The first computer-based systems were developed by following a bottom-up approach and were relatively unstructured. Users played minor roles in actual system development, leaving the work to the information specialists. As the analysts developed the systems, they produced a lengthy, complex narrative called a functional specification. The functional specification was difficult not only for users to understand but also for programmers to follow.

Structured Revolution The second phase of systems development can be labeled the structured revolution, and it spanned the period from the early 1970s to early 1980s. Once a firm successfully implemented its accounting systems, the demand for more complex and larger systems emerged. The increased complexity of newer systems demanded many more lines of code, which, in turn, increased the size of the development team manyfold. Coordination of the development effort and communication between team members became difficult, and the solution was seen as "structured" systems.

The first evidence of structured techniques came in the form of structured programming, which was followed by graphical analysis and design tools such as data flow diagrams, entity-relationship diagrams, structure charts, and Warnier-Orr diagrams. The main goals of these graphical tools were to allow a top-down approach to system development, to enhance communication, and to simplify the maintenance process.

Using the top-down approach, the system was first defined at a general, overview level, and then successive refinement occurred until the bottom, primitive-level functions were clearly defined. The primitive-level modules were as independent from one another as possible. This structure allowed maintenance programmers to make changes to a single, simple module without having to worry about creating errors in other modules.

The Toolkit Approach The structured approach stuck, and the next step was to automate the structured tools. This effort began during the early 1980s and continues to this day. The automated tools are expected to:

1. Ensure that the graphical tools are used in a consistent and standard way.
2. Increase analyst and programmer productivity.
3. Provide for an accessible repository of data definitions for the developing system.
4. Allow for rapid maintenance and system enhancement.

The name *toolkit* comes from the fact that the various automated tools are not integrated. They are merely a collection or bag of tools. The next evolutionary step was to integrate these tools into a comprehensive package.

CASE (Computer-Aided Software Engineering) The CASE approach amassed worldwide support in the late 1980s and continues to draw more attention than any topic in the systems field. CASE applies the formal methods of engineering to software design.

Tools exist to support various stages of software development, but they seldom are integrated and seldom cover the entire development cycle. A special category of CASE tools, called I-CASE (Integrated CASE), supports development from the initial strategic planning stages to the maintenance of the operational system.

I-CASE is the current state-of-the-art in software development. Students in the systems analysis and design course should understand that this approach will likely be the working environment for pursuing a career as a systems professional.

Most analysis and design textbooks up to this point have been constructed around the toolkit approach, using a middle-CASE software package to teach the tools. Middle-CASE means that support is restricted to the design phase in the system development life cycle. The beginning phases of business strategic planning, business area analysis, information systems strategic planning and architecture, and enterprise data modeling are neglected in these texts, with the tools being seen as ends in themselves. *Systems Analysis and Design: An*

Organizational Approach, on the other hand, teaches the full development life cycle, including the methodologies, the tools, and the underlying theoretical constructs.

This Text Satisfies All Three Needs

The three needs of recognizing end-user computing, organizational influence, and I-CASE are all met by *Systems Analysis and Design: An Organizational Approach.* After completing a course that utilizes this text, the graduate should be able to use state-of-the-art CASE tools to develop computer-based systems for users, and also provide assistance to users as they develop their own systems. Such computer-based systems would support the organization's strategic objectives.

The Text Meets Current and Future Needs

Shortly before work on the manuscript began, The Dryden Press conducted a mail survey of instructors of the systems analysis and design course. Responses were received from 652 instructors who provided insights as to current techniques and future needs. This valuable information formed the basis for the entire project. The result is a text that supports a wide variety of teaching environments, including both four-year and two-year schools, and both business and computer science departmental settings.

In addition to the survey, valuable suggestions were also received from reviewers who reviewed preliminary and refined versions of the manuscript. These reviewers include Kirk Arnett of Mississippi State University, Brother William Batt, FSC, of Manhattan College, Jane M. Carey of Arizona State University West, Carl Clavadetscher, Dakota State University, Joey George of the University of Arizona, Robert T. Keim of Arizona State University, Paul Licker of the University of Calgary, Nilakantan Nagarajan of the University of Florida, Julio C. Rivera, Ludwig Slusky of California State University, Los Angeles, and Maureen C. Thommes, Bimidji State University.

We responded to this important feedback and produced a text that we believe more closely meets the needs of the introductory systems analysis and design course than does any other text on the market.

Organization of the Text

The text consists of fifteen chapters, organized into six parts. There are also ten technical modules, and an ongoing case that evolves over ten scenarios.

All chapters are tied to state-of-the-art CASE technology by means of explanations that relate the technology to the chapter contents. In addition, each chapter contains an example of how CASE technology can be applied at that particular developmental phase.

Part One—Development of Computer-Based Information Systems This part consists of two chapters and a technical module. Chapter 1 provides an introduction to systems work and describes career opportunities. Chapter 2 describes the organizational framework within which modern systems work is performed. Technical Module A describes a high-level data modeling tool, entity-relationship diagrams.

Chapter 1 lays the important foundation for the study of systems work, regardless of the course approach. Chapter 2 is used when the instructor wants to provide the students with the organizational setting within which the work is performed. Technical Module A is positioned after Chapter 1 for a purpose. When the instructor wants to take a data modeling approach, the module can be assigned during the first week of class, and students can begin using the tool in a laboratory setting as early as the second week.

Part Two—Systems in Business This part views business operations in systems terms and consists of four chapters and one technical module. Chapter 3 presents fundamental systems concepts, and Chapter 4 applies the concepts to the organization, or firm. Chapter 5 explains the influence of the environment on the firm's activities. Chapter 6 explains how computer-based information systems are used to manage the firm as a system. Chapter 6 explains the major applications of computers in business—data processing, management information systems, decision support systems, office automation systems, and expert systems.

Technical Module B describes a popular tool for modeling the firm's processes—data flow diagrams.

Chapters 3, 4, and 5 are used when the course is to include a solid foundation of systems theory. Chapter 6 is used when the instructor wants the students to understand how the type of system influences systems work.

Part Three—Systems Methodologies A methodology is a recommended way of doing something, and there are quite a few in the systems field. The basis for all of the methodologies is the systems approach to problem solving, which is the topic of Chapter 7. Chapter 8 applies the systems approach to the development and use of a computer-based system in the form of methodologies such as the system life cycle, prototyping, and James Martin's Rapid Application Development (RAD). Technical Module C rounds out this part of the text by describing the tool used to model the firm's data in a detailed fashion, the data dictionary.

Instructors will include Chapter 7 when they want students to understand the theoretical basis for the methodologies. Most instructors will want to include Chapter 8 since it provides the most popular framework for studying systems development, the system life cycle.

Part Four—Systems Analysis This part of the text describes the process of understanding the users' needs so that they can be satisfied by a new or improved system. Chapter 9 is devoted to the first phase of the system life cycle, explaining how projects are planned and controlled. Chapter 10 is devoted to the second phase, systems analysis.

Technical Module D describes two types of network diagrams: Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT). These diagrams often form the basis for project planning and control. Technical Module E describes some popular approaches to economic justification that are commonly applied early in the system life cycle.

Part Five—Systems Design This part describes how a new or improved system is designed. Chapter 11 deals with the design process, and Chapters 12 and 13 address two special design issues. Chapter 12 describes how controls are built into systems to ensure that they perform as intended, and Chapter 13 addresses systems security.

The instructor will use Chapter 12 when the need for systems controls is to be emphasized and will use Chapter 13 when students are to understand how system designs can contribute to security.

Technical Module F describes graphical user interface (GUI) design, and Technical Module G describes flowcharting.

Part Six—Systems Implementation, Audit, and Maintenance This part completes the coverage of the system life cycle. Chapter 14 explains how systems are implemented, and Chapter 15 describes what happens after cutover—system use, audit, and maintenance. Two technical modules describe tools that can model systems processes in detail. Technical Module H explains structured English, and Technical Module I explains action diagrams. Technical Module J provides an example of a user manual that helps the user learn to use the new system.

Modular Design Facilitates Course Tailoring

The modular design of the text enables it to be used by the instructor to assemble exactly the right combination of chapters and technical modules. The *Instructor's Guide* describes six different combinations, and many more are possible.

Incorporation of Experiential Activity

The systems analysis and design course is perhaps the most difficult of the computer courses to teach because it is necessary to bring the real world of business into the classroom. This can be accomplished in several ways, and one of the most popular is the case problem.

The text includes two types of case problems. The **ongoing case** is based on the computer operations of the Harcourt Brace publishing firm in Orlando, Florida, and Bellmawr, New Jersey. As the students read the chapter material, they can see how it is applied in a real business setting. In addition, each chapter concludes with one or two short cases. These **end-of-chapter cases** enable the student to apply chapter material to solve a variety of problems.

In addition to the cases, the text also includes other opportunities for the students to put the material to use. Each chapter concludes with **end-of-chapter questions,** many of which require the student to apply the material in an innovative way. Also included are **end-of-chapter problems** and **end-of-technical module problems** that require the application of the tools and methodologies. In addition, there are **end-of-chapter discussion topics** that can focus the attention of the entire class on issues that are yet to be resolved.

A Grounding in the Literature of Systems

Modern systems work is a craft that has evolved over the years as a result of the contributions of many people. These people have sought to establish a written history and culture upon which to build the technology and to apply the methodologies and tools. The text is grounded to this rich systems literature.

Footnotes provide references to the literature where appropriate, and a **selected bibliography** is included at the end of each chapter, consisting of classics as well as leading-edge references. The footnotes and bibliography point to sources of additional information on selected topics as a means of gaining a full comprehension of modern systems work.

The Text Is a Systems Workshop

Because of its strengths in experiential activities and its solid grounding to the literature the text is a *self-contained workshop,* which enables the students to experience the challenges and rewards of systems work. Should the instructor prefer to stay within the bounds of the text, a successful course can be achieved. However, the instructor who wishes to incorporate additional learning tools has an opportunity to do that by using part or all of the complete package.

A Complete Package

Systems Analysis and Design: An Organizational Approach is supported by the following:

- An **Instructor's Guide** that contains suggestions for teaching the systems analysis and design course, a sample syllabus, and comprehensive lecture notes, keyed to textbook figures. Also included are solutions to all of the end-of-chapter questions and problems, as well as material that relates to the more controversial discussion topics, and suggested solutions to the cases.
- **Overhead transparencies** that consist of fifty two-color acetates of the most important textbook figures.
- **Transparency masters** that enable the instructor to use classroom projections of all textbook figures referred to in the lecture notes, plus many more.

- A **test bank** that contains true/false and multiple-choice exam questions, plus mini-quizzes for each chapter. The test bank is available in both a hardcopy and a computerized form.
- **Laboratory manuals** that provide the instructor with a choice of CASE tools for the students to use in applying the material from the text. One version supports the Visible Analyst from Visible Systems Corporation and is available for both DOS and Windows environments. Another supports the Information Engineering Workbench (IEW) from KnowledgeWare.

  The Dryden Press is also developing laboratory manuals to support the Information Engineering Facility (IEF) from Texas Instruments and Excelerator from Index Technology—for both DOS and Windows.

  The text is designed so that use of a lab manual can begin immediately following chapter 1. This organization enables the students to engage in hands-on, experiential activity *beginning in the second week,* stimulating student interest early on and maintaining it throughout the course.
- **Two videotapes** that support the Harcourt Brace ongoing case by providing a visual tour of both the Orlando computer project and the warehouse operation at Bellmawr. Students not only read about the project, but are able to see the actual facilities and meet the participants.

By taking advantage of this complete package, the instructor can assemble a course that captures the interest of the students and enables them to use the most modern systems technologies, methodologies, and tools.

A Team Effort

I use the term *we* rather than *I*. Some students find this strange since I am the sole author. I follow the *we* strategy mainly so that students will understand that they are a part of the effort. I like to think that the author, the instructor, and the students all follow the same path together.

At the same time, I also recognize that I do not do all of the work. The writing and publishing of a textbook is a team effort, and one of the rewards is the opportunity to work with the many fine people that a publisher such as Dryden has to offer. My main contact at Dryden has been Richard Bonacci, my editor. Richard has gone that extra mile to ensure that we produce a book that meets the needs of the instructors who teach the systems course and the students who take it. Also playing key roles at Dryden were Stacey Fry, Developmental Editor, and Michele Tomiak, Project Editor. I also recognize the support of Kevin Cottingim and Scott Timian, Marketing Managers, who guided us through some rough spots and planned the marketing campaign. From my very first introduction to Dryden, I was impressed with everyone's professionalism. That impression has persisted.

Acknowledgments

Many people outside of Dryden have contributed to this effort as well. Some are from academia, and some from industry.

Much of my academic help comes from my colleagues at Texas A & M University. We have one of the top MIS faculties in the country, and professors Joobin Choobineh, Michael Chung, Jim Courtney, George Fowler, Bill Fuerst, David Paradice, Arun Sen, Marietta Tretter, and Ajay Vinze were always ready to help when needed. Two doctoral students, Hae-ching Chang and Choong Kim, and a masters student, Paul Davis, contributed as well.

I am especially indebted to Joobin Choobineh for preparing Technical Module A on entity-relationship diagrams. Jane Carey of Arizona State University West also contributed

Technical Module I on action diagrams and wrote the CASE segments that appear in each chapter.

Persons from industry have also provided important material. I am indebted to the management of Harcourt Brace for opening the doors of their Orlando and Bellmawr facilities to me for the purpose of developing the ongoing case. I specifically recognize Mark Arak, Neil Aronow, Michael P. Banks, Michael W. Byrnes, Bob Evanson, Alan Fox, Anne Hogan, Ann Johnston, Jenny R. Jolinski, Ira Lerner, Sal Leonetti, Suzanne McNulla, David J Mattson, John Misuira, Cathy Oliva, William Presby, David Renk, and Michael J. Tolen. I also acknowledge the help from Debbie Sandoval of Comshare; Susan Benjamin of Rollerblades; and John Hoffman of Skill Dynamics™, an IBM company.

In writing the chapter on systems security, I was assisted greatly by Dahl Gerbrick and Daniel Faigin of ACM's Special Interest Group on Security, Audit, and Control—SIGSAC. Dahl put me in contact with three of the nation's top security experts—Ken Cutler of American Express, Thomas C. Jones of Phoenix-based Lemcom Systems, Inc., and Daniel E. White of Ernst & Young.

I also want to acknowledge the role played by my systems analysis and design students. Much of the material in this text was developed in class, with valuable suggestions and insights coming from the students. I especially recognize the systems class of the Spring 1992 semester, which used the initial version of the text manuscript. In that class, Melissa Woodlan and Greg Liddle contributed the basic structure of the office automation model described in Chapter 6. Students in my Spring 1993 MIS classes also contributed. Darren Stokes and Tom Kozelsky provided examples of graphical user interfaces, and Cynthia Caldarola provided the user manual that appears in Technical Module J.

I hope that you, the instructor and students, like the book. Please let us know. Our plan is that the book will be around for a long time and will get better with each edition. We can do that only with your help.

I have enjoyed working on the book and have learned a lot in the process. May you experience the same rewards in using it in class and applying it in your job.

Raymond McLeod, Jr.
College Station, Texas
January, 1994

CONTENTS