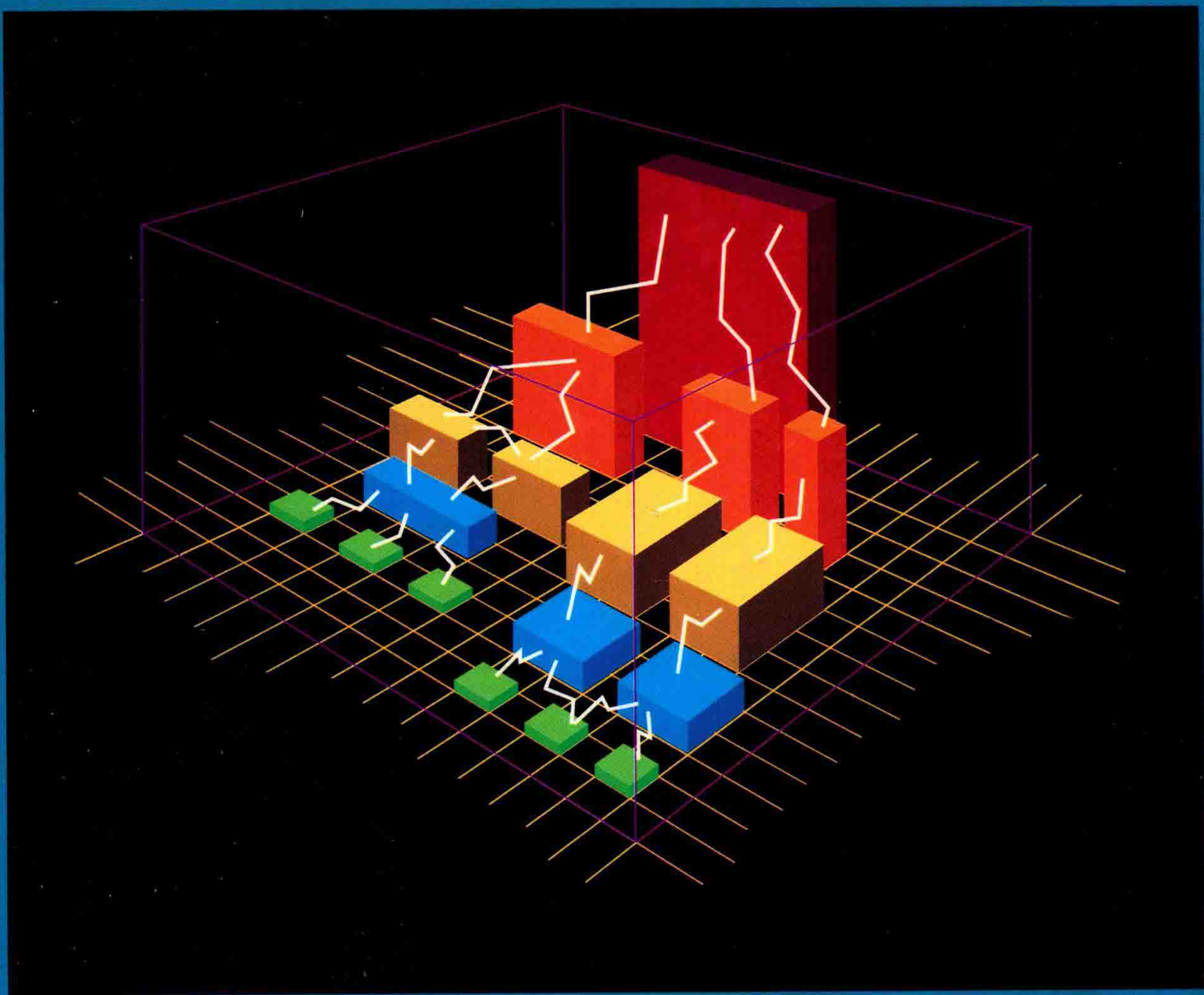
JAMES C. WETHERBE



Systems Analysis and Design:

Traditional, Structured, and Advanced Concepts and Techniques

Second Edition

Systems Analysis and Design

Traditional, Structured, and Advanced Concepts and Techniques

Second Edition

JAMES C. WETHERBE

Professor and Director
Management Information Systems Research Center
School of Management
University of Minnesota

West Publishing Company

St Paul . New York . Los Angeles . San Francisco

50 West Kellogg Boulevard
P.O. Box 3526
St. Paul, Minnesota 55165

Printed in the United States of America

Library of Congress Cataloging in Publication Data

Wetherbe, James C.
Systems analysis and design.

Rev. ed. of: Systems analysis for computer-based information systems. 1979.
Bibliography: p.
Includes index.
1. Management information systems. 2. System analysis. I. Wetherbe, James C. Systems analysis for computer-based information systems. II. Title.
T58.6.W47 1984 658.4'03'2 83-23461
ISBN 0-314-77858-6
ISBN 0-314-77859-4 (international ed.)

Systems Analysis and Design

Second Edition

Preface

It is a humbling experience to revise a book only four years after it was first published. Though I have since written four other books, this is my first revision of a book. In revising, it became glaringly apparent that the book had become out of date. Indeed, the year this book was published many aspects of it were already out of date. This shows how difficult it is to keep current information available for teaching our future systems analysts.

In preparing to revise the book, I have reviewed the curriculum recommendations provided by the Data Processing Management Association and the Association for Computing Machinery and have found that these also have become dated. Thus, in revising this book, I am sensitive to the curriculum recommendations for the systems analysis course for both curriculum models; however, I have added material above and beyond what was included in those recommendations. For example, I have gone beyond traditional and structured techniques and added advanced systems development techniques, including heuristic development and prototyping. I have also focused on some of the newer technologies such as fourth and fifth generation languages, personal computers, information centers, end-user computing, artificial intelligence, and expert systems. I have incorporated the impact of these technologies into the systems development process. Also, I have added a very useful planning concept in the last section of the book. This can provide the structure needed to develop individual applications within an integrated framework.

Overall, this book has a stronger emphasis on information and business analysis than the first edition. This emphasis is consistent with the trend toward increasing the ability of systems analysts to understand and solve the problems of user management.

One commitment I have made in revising the book is to publish a newsletter containing new articles that are pertinent to it. These should be used as a supplement to the book. This newsletter will be available upon request through me at the University of Minnesota to those using the book for educational purposes. This will allow those using the book to be aware of aspects of it that have become dated and to add new material from these articles to their curriculum. The case book that goes with this text has also been updated to reflect the new approaches in this book. Also, a set of transparency masters for presenting the material in the classroom is available to instructors.

The interesting thing I find in revising the book is that those aspects of it that were conceptual and theoretical have not changed much. This is not too surprising, as theories and concepts tend to be stable; it is the techniques that are continually changing. Because of this, the architecture of this book is very similar to that of the first edition.

The first section of the book, Chapters 1–4, focuses on conceptual foundations. These set the stage for learning those specific techniques of systems development covered in Chapters 6–10. Chapters 11–14 focus on strategic, administrative, and other higher-level techniques. These high-level techniques include the planning and developing of strategic information, information architectures, resource allocations, decision-support systems, and future considerations. In the first book, I viewed database management systems as being advanced; now I see that they are available in most organizations. I therefore consider it a routine rather than an advanced topic and restructured the text accordingly.

Use of Text

The material in this text is designed for the student who has previously completed, or is now taking, a course involving file-oriented, procedural programming language such as COBOL, or PL/1. It has been successfully used by both undergraduate and graduate students.

Acknowledgements

In conclusion, I wish to acknowledge the contributions of several individuals. I am especially indebted to V. Thomas Dock, University of Southern California, who encouraged me to write this book and served as consulting editor. The reviewers whose helpful suggestions are reflected throughout the text are: Marily Bohl, IBM; Thomas I. M. Ho, Purdue University; Roger Hayen, University of Nebraska; Steven Alter, University of Southern California; Richard Scamell, University of Houston; M. Bond Wetherbe, NAVDAC; Tracy Galun, University of Minnesota; Charles Davis, MIS, Occidental Petroleum Corporation; Hank Lautenbach, Idaho State University; Charles Paddock, Arizona State University; Phillip Judd, University of Houston; Everald E. Mills, Wichita State University; Robert T. Keim, Arizona State University, and Kenneth W. Veatch, San Antonio College.

I offer sincere thanks to my patient, tolerant, and forgiving support staff: Katherine A. Cooper, Nancy L. Draper, Leslie Maggi, and Susan M. Scanlan.

Finally, I acknowledge the efforts and support of my wife, Smoky, who is a supportive friend and colleague in my professional efforts.

As I acknowledge the efforts of all who have contributed, I also assume full responsibility for any inadequacies or discrepancies in the text.

JAMES C. WETHERBE
Minneapolis, Minnesota
1984

Contents

SECTION I

Conceptual Foundations

CHAPTER 1

Overview of Systems Development

INTRODUCTION

Organizations are under greater pressure than ever before to increase productivity. Increasing productivity has become a key to remaining competitive and viable in today's industrialized society. Yet the very resources that organizations need to remain competitive cost more. People, equipment, facilities, capital, and energy all cost more and therefore increase the cost of producing goods and services. The only exception to this (and a glaring exception it is) is computer technology. Computers continue to drop in cost at a phenomenal rate. An analogy is often made between the computer industry and the automotive industry. If the automotive industry had had the same breakthroughs that the computer industry has had during the past thirty years, then today you could buy a new Cadillac for less than a quarter. No other industry in our society is offering that much increase in cost performance.

Accordingly, computer technology and information systems are the best game in town with which to increase organizational productivity. Those people who know how to apply this powerful technology hold the keys to the future.

Organizations are just now realizing that we are truly moving into the information society. Information is now viewed as a strategic corporate resource. Most Fortune 500 companies have identified computers and information systems as one of their top strategic issues for remaining competitive in the future.

As a resource, information has characteristics that organizations have never really dealt with before. For one thing, unlike other resources such as fuel and real estate, information is not a scarce resource. It also is a nondepleting resource. Perhaps its most unusual characteristic is that it is a resource that you can have, give to someone else, and yet still hold. You can't do that with a barrel of oil or with investment money.

Indeed, it is this characteristic of being able to share information that makes it such a powerful resource. Information is the glue that holds organizations together. It is the resource that allows the left hand to know what the right hand is doing. It is the resource that can avoid the incredible waste that occurs when an organization is poorly coordinated due to poorly informed managers.

It is these characteristics of computers and information, and the needs for organizations to increase organizational productivity, that makes the study of systems analysis and design such a worthwhile endeavor. This is true whether someone aspires to be a systems analyst or to hold any other staff or management position within an organization.

Irrespective of where you ultimately end up working, computers and information systems are going to be a key dimension of what you do. No matter what management or staff position you hold, the most important thing you will be doing is making decisions. Other than your intellect, the most important resource you will need to do your job well will be information.

To set the stage for studying material in this book, certain key foundations must be established. These will be discussed next.

THEORY VS. TECHNIQUE

First, it is very important to understand the difference between theory and technique. Theory is really nothing more than the notions or ideas about the way things work or the way things are. Techniques are just different methods or approaches to performing different tasks. Ideally, techniques should be based upon certain theories.

People often get confused, and debate the issue of theory vs. technique. Some people will argue that the key is to have good technique, while others argue the key is to have good theory. The two concepts, however, complement each other and should be used accordingly. Someone who knows technique without theory, is potentially a dangerous person. Such a person uses various techniques or methods without really understanding the reasons behind them. Anytime someone uses a particular technique without knowing the reason or the theory behind it, that person is apt to use the technique when it is inappropriate. For example, consider the young man who would always cut a roast in half before he would cook it. His wife, observing him do this, asked him, "Why do you cut the roast in half before you cook it?" The husband replied, "I don't know. My mother always did, so I always have." After awhile the young man became curious as to why it had been a family tradition to cut the roast, so he asked his mother about it. His mother explained that she had always done it because *her* mother had always cut the roast in half before cooking it. Their curiosity led them to question the grandmother as to why the roast was always cut in half before it was cooked. The grandmother had a simple explanation—the reason she cut her roast in half was because her pan was so small it was the only way that she could fit the roast into the pan. Though the grandmother had a reason for using the roast cutting technique, no one else did.

We often find people in different professions, including systems analysts, using techniques that are outdated or no longer appropriate. This generally happens when they don't understand the theory behind the technique.

It is not enough, however, just to know theory. Someone who knows theory and no technique is useless. What is needed is an understanding of both the theory behind something, and the techniques that can be used to solve problems based upon that theory.

Ignorance, Knowledge, and Counterknowledge

Understanding the importance of knowing both theory and technique, leads to a discussion of ignorance, knowledge, and counterknowledge. Ignorance is when someone *does not* know how to do something—perhaps a technique for doing something. Knowledge is when someone *does* know how to do something or a technique for doing something. Counterknowledge is when someone thinks they know how to do something or the best technique for doing it, but they are wrong. They might be wrong because they are using a technique in the wrong situation, or they might be wrong because the technique is no longer the best technique for solving a particular problem. You will generally find people guilty of counterknowledge when they really do not understand the theory behind the technique they are using.

A common criticism made of the educational process is that it tends to teach students how to do exercises rather than to solve problems. For example, we teach students how to write a computer program, how to solve a mathematical equation, or how to do a financial analysis. The students, however, often do not know when it is appropriate to *use* any of these techniques. This is similar to the story of the student who is taught how to use a screwdriver—the screwdriver being symbolic of some technique such as computer programming or linear programming or net-present-value analysis. Upon being employed, the student is excited about applying the screwdriver to screws that need to be tightened. The new employee goes about the organization looking for screws to tighten and tightens them. Eventually, however, there are no screws left to tighten. Looking for new opportunities to apply the skill of using a screwdriver, the new employee gets out a hack saw and starts filing slots into the heads of nails and then getting out the screwdriver and using it upon the nails. As silly as this story may seem, it is unfortunately representative of the application of an inappropriate technique to the solution of a problem.

It is very important when solving problems to understand what the real problem is, what theory is relevant to the problem, and what technique, or techniques best fit the situation.

Methodology Lags Technology

A key aspect of selecting the right technique or methodology to solve a problem is being aware of the most current techniques or methodologies. In all fields, techniques and methodologies tend to lag behind technology. In other words, a set a techniques or methodologies are developed based upon technology that is available at a point in time. As time passes, new technologies are introduced, but people tend to use the old methodologies that were developed for a previous generation of technology. This often results in considerable loss in problem-solving potential and productivity.

For example, let's move out of a computer and information systems context, and consider the Revolutionary War. During the Revolutionary War, the British used a methodology of warfare in which soldiers in festively colored uniforms lined up shoulder to shoulder several deep, and marched through open fields. The Revolutionaries, on the other hand, wore drab clothing that tended to blend in with the rocks and the trees that they hid behind. Thus they were able to have a strong competitive advantage in the warfare. Now it might seem a little peculiar that the British were willing to wear bright red jackets with white X's across the front and march into open fields and open gunfire. Why, in combat, would they use a methodology that clearly was not in their best interest? The problem was that their methodology lagged their technology. The method of lining up in a formation of thousands and marching at the enemy was born during the sword-and-shield technological era. The idea of creating an awesome looking army and marching at the enemy evidently was quite effective during that era. The introduction of rifle technology, however, made such a methodology fatally obsolete. The British literally offered target practice to the revolutionaries and suffered as a result of it.

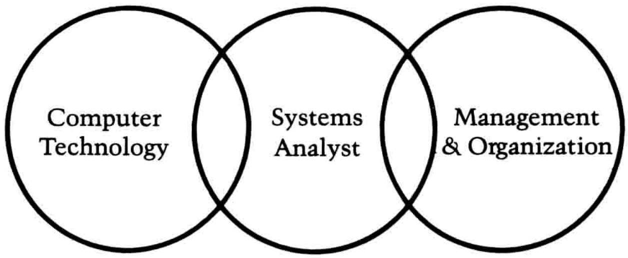
If we proceed to look at the Civil War, we note the phenomena of counterknowledge. As military tactics became more "sophisticated" in America, the method of lining up soldiers and marching at the enemy became the tactic of American soldiers. Consequently, during the Civil War, both Union and Confederate soldiers marched at each other openly. The results were devastating. The fatalities of the Civil War were greater than all Americans killed in both World War I and World War II. It wasn't until the twentieth century that the ideas of wearing camouflage clothing, of digging foxholes, and other tactics more appropriate for dealing with rifle technology were developed.

Similarly, many aspects of what we are now doing in information systems development are quite obsolete, given the technologies available. In other words, methodology is lagging technology. One way to become sensitive to this phenomena, and therefore have the opportunity to minimize it, is to be sensitive to what the theories and techniques are for systems development. Theories tend not to change as rapidly as do methodologies or techniques. For example, one theory pointed out in this book is that managers do not know what information they need. Though that theory has been around since the 1960s, the techniques used to deal with it have changed significantly. Consequently, a good understanding of theory can assist one in reviewing and revising methodologies to catch up with technologies.

Throughout the book we will emphasize what is theory and what is technique, and try to point out how we are catching up with technology.

WHAT MAKES FOR A GOOD SYSTEMS ANALYST

Information systems analysts are people who are involved in analyzing, designing, implementing, and evaluating computer-based information systems to support the decision making and operations of an organization.[1] They are ostensibly a boundary spanner between computers and management as illustrated below:



As illustrated in the circle graph, systems analysts must understand the technology, the organization, and the skills of their trade. To refine this further, the key ingredients that make for a good systems analyst are:

1. Understanding of, and commitment to, the organization.
2. People skills
3. Conceptual skills
4. Technical skills

Each ingredient is discussed in the following text.

Understanding of, and Commitment to, the Organization

One of the biggest complaints made of system analysts is that they don't understand, and are not committed to, the organizations in which they work. The typical complaint is that systems people are more interested in the technology they work with than in the organization that uses it.

This problem can be illustrated by an example. I was working with a medium-sized department store chain that was installing its first computer. In the process this chain was switching from electromechanical

---

[1]This discussion is based on a chapter from James C. Wetherbe, *Executive Guide to Computer-Based Information Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1983), pp. 156–60.