INFORMATION ENGINEERING FOR THE PRACTITIONER

Putting Theory into Practice

W. H. Inmon

INFORMATION ENGINEERING FOR THE PRACTITIONER

Putting Theory into Practice

W. H. Inmon

YOURDON PRESS
PRENTICE HALL BUILDING
Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

Inmon, William H.
Information engineering for the practitioner.

(Yourdon Press computing series)
Bibliography: p.
Includes index.
1. System design. 2. Data base management. I. Title.
II. Series.
QA76.9.S88154 1989 005.74 88-2507
ISBN 0-13-464579-0

Editorial/production supervision: *Raeia Maes*
Cover design: *Wanda Lubelska*
Manufacturing buyer: *Mary Ann Gloriande*

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-464579-0

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, New Delhi
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

YOURDON PRESS COMPUTING SERIES
Ed Yourdon, *Advisor*

BENTON AND WEEKES Program It Right: A Structured Method in BASIC
BLOCK The Politics of Projects
BODDIE Crunch Mode: Building Effective Systems on a Tight Schedule
BRILL Building Controls Into Structured Systems
BRILL Techniques of EDP Project Management: A Book of Readings
CONSTANTINE AND YOURDON Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design
DE MARCO Concise Notes on Software Engineering
DE MARCO Controlling Software Projects: Management, Measurement, and Estimates
DE MARCO Structured Analysis and System Specification
DICKINSON Developing Structured Systems: A Methodology Using Structured Techniques
FLAVIN Fundamental Concepts in Information Modeling
FRANTZEN AND MC EVOY A Game Plan for Systems Development: Strategy and Steps for Designing Your Own System
HANSEN Up and Running: A Case Study of Successful Systems Development
INMON Information Engineering for the Practitioner: Putting Theory Into Practice
KELLER Expert Systems Technology: Development and Application
KELLER The Practice of Structured Analysis: Exploding Myths
KING Creating Effective Software: Computer Program Design Using the Jackson Method
KING Current Practices in Software Development: A Guide to Successful Systems
MAC DONALD Intuition to Implementation: Communicating About Systems Toward a Language of Structure in Data Processing System Development
MC MENAMIN AND PALMER Essential Systems Analysis
ORR Structured Systems Development
PAGE-JONES The Practical Guide to Structured Systems Design
PETERS Software Design: Methods and Techniques
ROESKE The Data Factory: Data Center Operations and Systems Development
SCHLAER AND MELLOR Object-Oriented Systems Analysis: Modeling the World in Data
SEMPREVIO Teams in Information Systems Development
THOMSETT People and Project Management
WARD Systems Development Without Pain: A User's Guide to Modeling Organizational Patterns
WARD AND MELLOR Structured Development for Real-Time Systems, Volumes I, II, and III
WEAVER Using the Structured Techniques: A Case Study
WEINBERG Structured Analysis
WELLS A Structured Approach to Building Programs: BASIC
WELLS A Structured Approach to Building Programs: COBOL
WELLS A Structured Approach to Building Programs: Pascal
YOURDON Classics in Software Engineering
YOURDON Design of On-Line Computer Systems
YOURDON, LISTER, GANE, AND SARSON Learning to Program in Structured Cobol, Parts 1 and 2
YOURDON Managing Structured Techniques, 3/E
YOURDON Managing the System Life Cycle, 2/E
YOURDON Structured Walkthroughs, 2/E
YOURDON Techniques of Program Structure and Design
YOURDON Writing of the Revolution: Selected Readings on Software Engineering
ZAHN C Notes: A Guide to the C Programming

PREFACE

The need for an engineered, architected approach for information systems has been recognized for a long time. And in bits and pieces an engineering approach has evolved. The theoreticians have contributed some notions as to how to do high-level data modeling. The technicians have contributed some notions about how to accommodate mundane needs such as data-base performance. The system developers have contributed some ideas concerning the importance of recognition of commonality and uniqueness of data and processing. Management has recognized the economics of consolidation that are possible, but, truthfully, very few (if any) attempts have been made to bring together *all* the necessary components needed to build a complete, unified, practical, theoretically sound, economical, and *proven* approach to information engineering.

Information engineering spans the full spectrum of engineering activities, from the very earliest conceptual decisions to the detailed design decisions in a comprehensive and integrated manner. Furthermore, the considerations and techniques brought to the attention of the reader have been distilled from many actual experiences. The effective applications of theory have been culled from ineffective approaches and have been put into the context of a workable, real, and efficient approach to the achievement of information engineering.

This book is for system developers, data administrators, data-base administrators, programmers, business students interested in the application of computer technology, students of computer science, and any other interested party who is concerned with an engineered approach to information processing and systems. The book is for the reader with a working knowledge of the different phases of systems development. For the first-time reader into these subjects, there are a plethora

of books on the subjects of structured development, data-base design, and project management. The book will be of special interest to theorists who are concerned with the actual results achieved by the body of theory that has grown up around information engineering over the years. Also, system developers wishing to apply rigor to the design and development process will find the book full of practical insight.

Some of the topics covered in the book are fairly well known; some are not. The major topical highlights of the book include the following:

- Development of a three-level conceptual data model.
- How data and process models relate.
- Practical and efficient development of a corporate ERD (entity relationship diagram).
- Development of a high-level process model.
- Development of a mid-level data model, the dis (or data item set) level. The dis provides the transition between the high-level corporate data model and the ensuing physical model. The proper construction of the dis is the key to the recognition and management of commonality and uniqueness of data.
- Construction of the physical model from the mid-level model.
- Concepts relating to the conformance to the architecture throughout the development life cycle.
- Clear and deliberate separation of primitive and derived data and processing.
- Development of a physical architecture that includes operational, atomic, departmental, and individual processing and data, with a clear interface between the different levels.
- Recursive data structures.
- Time-dependent data structures.
- Dss/end user computing data structures and architecture.
- Software support of the architecture.
- Distributed data structures and processing.
- Data structures for the parallel environment.
- Archival data structures.
- Restructuring of data structures.
- Data structures for high performance.
- Data structures for high availability.
- Main storage data structures.

Throughout the book emphasis is on effectiveness and do-ability. The author has witnessed several very good information engineering efforts and several dismal failures. The book, in the final analysis, is a compendium of what works and why, and what doesn't work and why.

After reading the book, the reader will be prepared to face the development effort with a discipline borne of a blend of theory and how the theory is best applied.

ACKNOWLEDGMENTS

The author wishes to acknowledge the contributions of several people, who played either active or passive roles in the development of this book, but who in every case contributed substantially, either in concept or in material.

Bob Jacobucci of Fireman's Fund gave the author the insight as to what is required to implement an architecture. To this date, the author looks to Bob and Fireman's Fund as a very good example of what is required to implement an architecture.

Newell Hall, Doug McQuilken, Pat Keck, John Eckert, Rachel Markowitz, and many others at Aetna Life and Casualty kindly shared many profound ideas about a three-level architecture. The author sincerely hopes that he has contributed in kind.

Allan Gibson, Phil Smith, Craig Venator, and Wayne Wilhelm of Coopers & Lybrand provided the author with the opportunity to exercise his ideas in practice and challenged the author to make the concepts actually work. For this opportunity, the author is forever indebted.

Dave Menke, Paul Dwyer, Jerry Scherkenbach, Bruce Ziebell, Ron Kundinger, and Harvey Lippow of First Bank of Wisconsin contributed valuable ideas on organizing the output of user view sessions, preparing for the high-level and mid-level modeling activities.

The many attendees at the author's seminars and lectures have always provided fresh insights and perspectives, which, quite frankly, the author would never have experienced. For the many questions, even the really off the wall ones, that the author has been asked over the years, thank you.

Barbara Graham of Coopers & Lybrand has, from the very start, been an asset, both as a colleague and as a student of information engineering. Barbara was involved in some early work done by the author and remains a very valued friend and professional acquaintance.

Gary Dodge of Coopers & Lybrand has been a friend and confidant, always quick with constructive criticism, which is always appreciated.

Thanks to Carol Snyder of American Management Systems for her word processing support and for catching many errors that otherwise would have made the book awkward.

Thanks to Maris Dukas and Dr. Jerrold Grochow for the software examples from American Management Systems. And thanks to Bill Clarke of Index Technology for software examples in support of data and process modeling.

To all of the many people who have contributed in large or small part, the author wishes to express a sincere thank you.

W. H. Inmon

CONTENTS

SECTION I

AN INTRODUCTION TO INFORMATION ENGINEERING

A good plan today is better
than a perfect plan tomorrow.

General George Patton

CHAPTER 1

DATA-DRIVEN DESIGN: FROM MASTER FILES TO AN ARCHITECTURE

A HISTORICAL OVERVIEW

In the early dawn of system development and design, in the late 1950s and early 1960s, the focus of the developer was almost entirely on making technology work. The challenges of merely getting technology up and running were such that very little attention was given to the way systems were being built. The issues of productivity and quality were given little or no attention in the face of the huge obstacles posed by merely making technology perform.

The attention of the developer was focused on algorithms, code, and the initial break-in of technology. The pace of work, the pressure to build systems, and the challenges of harnessing technology were such that very little energy remained for anything other than the satisfaction of immediate goals.

In this early environment, the perspective taken by the system developer, out of necessity, was that of limiting the many variables that must be managed to create a system. By shutting out many broad considerations, the developer took what can be termed a "local" perspective. The local perspective is entirely defensible given the complexity and pressure facing the developer in the earliest days of system development.

Born out of the local perspective was the magnetic tape-oriented "master file." The limitations caused directly and indirectly by the master file approach to systems are legendary. In the master file environment, data is accessible in only a sequential manner. The typically limited number of magnetic tape drives forced the system developer into producing many separate master files, each of which contained a large amount of redundant data. The inability of a single master file to be accessed by more than a single program at a time forced the conglomeration of many different