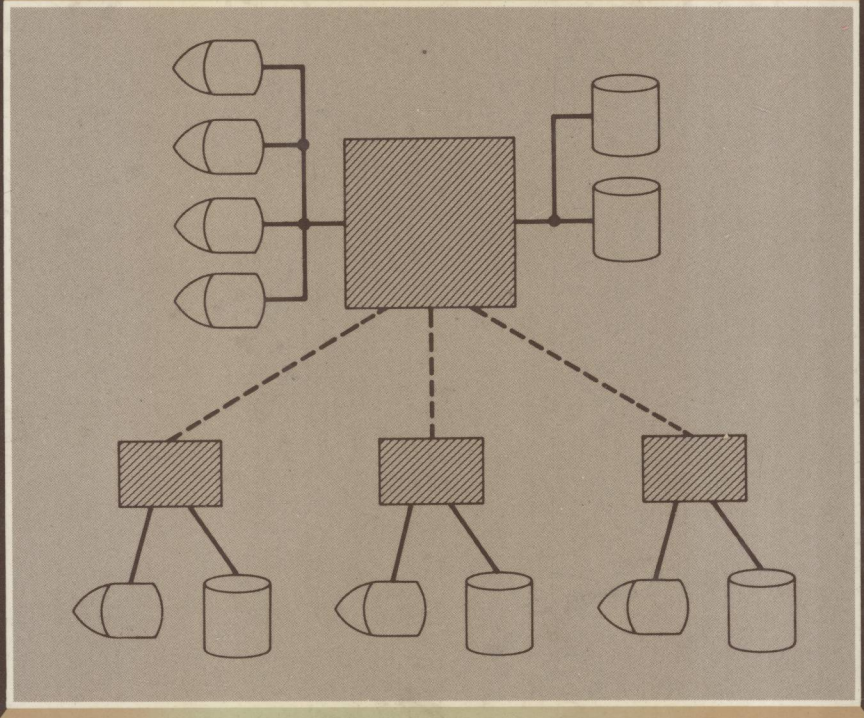
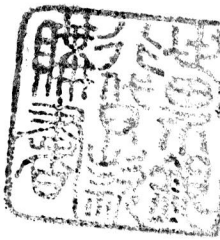
W.H. Inmon

INTEGRATING DATA PROCESSING SYSTEMS

In Theory and in Practice



PRENTICE-HALL SERIES
IN DATA PROCESSING MANAGEMENT
Leonard Krauss, *Editor*



INTEGRATING DATA PROCESSING SYSTEMS

In Theory and In Practice

W. H. INMON

PRENTICE-HALL, INC. / Englewood Cliffs, NJ 07632

Library of Congress Cataloging in Publication Data

Inmon, William H.
Integrating data processing systems.

Bibliography: p.
Includes index.
1. Electronic data processing. 2. System design.

I. Title.
QA76.I485 1984 001.64'404 83-16058
ISBN 0-13-468991-7

Production/interior design: Nicholas Romanelli
Cover design: Edsal Enterprises
Manufacturing buyer: Gordon Osbourne

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-468991-7

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janiero*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, Wellington, *New Zealand*

INTEGRATING DATA PROCESSING SYSTEMS

In Theory and In practice

for my sister, June

PREFACE

The major applications of commercial data processing at present are large operational systems used by banks, in manufacturing processes, for airline reservations, and so forth. Such systems are the heart of day-to-day operations, providing a flow of information that is used to manage the direction of each enterprise.

Many companies have existed for years prior to the advent of data processing systems. Automation has created an opportunity for implementing new capabilities and competitive advantages. Too often such systems have been created piecemeal, one step at a time. Sometimes such fragmented construction of systems has occurred because of a lack of planning; on other occasions fragmented development has occurred for purely political reasons, and in some instances there may have been technical factors that contributed to such development. In any event such piecemeal system development is, in the long-term, a detriment to any enterprise.

The result of fragmented development is a chaotic environment where systems or their components are designed many times and where system components that should be compatible are not. Multiple systems require multiple development efforts and, worse, require multiple maintenance efforts. Eventually productivity of a data processing department progressively diminishes. If the era of information engineering is ever to be realized, the approaches to system development *must* be changed from haphazard design and development practices to orderly, unified approaches—in effect, the wheel should be invented once and maintained once. From the broad perspective of an entire company (not from the traditional limited perspective of its political and departmental divisions), the economies

are so profound that the company *cannot afford* an installation that does *not* represent an orderly approach to system development. Unification of a system *mandates* that it be integrated. However, changes in approaches and practices, especially those as solidly entrenched in data processing installations today, often meet much resistance. The first step toward achieving the integrated environment is understanding integration and the perspective it entails.

Systems integration in data processing is a widely discussed concept, but one of the most elusive to achieve. Exactly what does integration mean in terms of data processing? Integration, in a word, implies unification—a single system image. When a system is integrated the user views the system (its data and its processes) in a consistent fashion wherever and however the system is used. The integrated system yields consistent and accurate responses.

Integration of a system of data processing also implies that two major considerations have been approached properly and in unison. These two factors are the physical and logical design of the system. Quite naturally the physical factors of a system deal with its physical components (operating systems, network monitors, hardware) and how they are organized (centralized, decentralized, and so forth). The logical considerations concern the organization of processing and data. Of major concern is how the initial system design is planned. Of equal importance is how ongoing changes are made to the system so that it remains integrated.

Topics of special interest in this book are:

—integration and the physical components of a system
—integration and centralized and decentralized systems
—the spectrum of integration
—the chandelier perspective of integrated systems
—initial integration of systems
—ongoing integration of systems
—integration and system ownership
—integration and system changes
—integration and large accumulation of data.

There are many concepts in this book that cannot be found elsewhere in the literature because integration of systems at the corporate level is new and there is a limited number of cases from which this material can be drawn. Within the literature on data processing

some of the subjects considered in this book are discussed theoretically, but nowhere is there a perspective of *all* the appropriate subjects that brings together and assimilates the material. Thus the reader is exposed here to many new ideas as well as to an organization of existing ideas.

The concept of system integration has existed since the debut of computers. Advantages of system integration are most easily understood in the negative, that is, in terms of an unintegrated environment. Unintegrated systems conspicuously waste expensive resources such as manpower and computer power. Duplication of effort in unintegrated systems, their maintenance, and their synchronization are obviously only the tip of the iceberg. When systems are integrated such expensive activities disappear or are minimized. The lack of system integration and of an underlying appreciation of the issues involved are great obstacles to success in data processing today.

The benefits of integration are obvious: efficient use of manpower, maximum use of computing facilities, and synchronization of data. But how to achieve integration is anything but obvious. An integrated environment is one where individuals invent the wheel only once, so to speak. For many reasons (none of them good) the wheel is constantly being reinvented in the data processing environment and, furthermore, developers are quite content with such reinvention. This book explains what system integration is (and is not), why integration is so difficult, and what is required to create and maintain an integrated environment.

System integration in the data processing environment has many diverse aspects. To achieve integration one must come to grips with such various issues as hardware, software, development practices, coding practices, and management practices and attitudes.

Because of its many facets, integration—when it is achieved—is accomplished by degrees. We attempt in this book to provide a framework for understanding the issues of integration and achieving an integrated environment. It is written for managers, system developers, designers, programmers, students of data processing, and maintenance personnel. Such readers will find the issues of integration meaningful to them. The book is written at a conceptual level, with illustrations discussed in detail. Our intent is to describe the issues of integration so that the patterns are recognized as they will appear in the reader's environment. However, discussions are not so theoretical that the reader will be unable to apply principles.

After studying this book the reader should be able to discern an integrated and an unintegrated environment, understand the advantages of integration, have a conceptual basis for achieving integration, and apply the concepts to his or her own environment.

In many ways this book represents but an introduction to the many facets of system integration. The questions that appear at the end of each chapter are designed to further understanding of the issues that are presented. Attempting to find solutions to the questions will lead the reader into explorations that will yield great insight into a complicated subject. In some respects the questions represent the author's desires: had the book been many times longer, each of the questions could have been explored in depth. Unfortunately limitations of space prevent such explorations, but the questions do point along a path that is sometimes uncharted, and in every instance worthwhile traveling.

Several individuals have been helpful during the writing of this book. Jeanne Friedman made some useful contributions for the chapters on abstraction and a data dictionary. Greg Estep offered some significant comments on data views and process views. Cheryl Estep offered good cheer and encouragement throughout. There are two other people whose contributions were notable. Throughout this book, directly or indirectly, Bob Brown's influence appears. Without his pioneering efforts it is possible that this book would never have been written. And our special thanks to Thomas Hawkes, who led the effort while integration was being transformed from an idea to a viable entity. The author extends his appreciation and gratitude to each of these individuals. And finally, the author thanks Coopers & Lybrand for their continuing support.

W. H. Inmon

INTEGRATING DATA PROCESSING SYSTEMS

In Theory and In practice

CONTENTS

Section I

What Integration Is (and Is Not)

Don't try to teach a pig to sing:
it annoys the pig and wastes your time.
American homily, ca. 1982

Chapter 1

The Framework of Integration

The task of integrating data processing systems can be broken into two major subtasks: the task of building a framework in which integration can occur, and the task of internally constructing the systems which will run in that framework. Both tasks are absolutely essential to the achievement of the integrated environment, but they involve surprisingly different sets of considerations.

The framework within which integrated systems are built is made up of various components. Most organizations purchase those components that make up the framework. Purchasing components of the framework is attractive because of the size and complexity of the components, their availability, and the ongoing support that is available in the eventuality of a failure or problem with a component.

On the other hand, in acquiring systems that will fit within an integrated framework, organizations choose to build most or all of their own systems. In-house construction of applications is desirable because of the uniqueness of the systems, the security that can be achieved, and the ability to respond in a timely fashion to changes in the nature of the system. In short, organizations feel comfortable in controlling their own applications.