Building the Data Warehouse, Fourth Edition

W. H. Inmon

WILEY
Wiley Publishing, Inc.

Building the Data Warehouse, Fourth Edition

Building the Data Warehouse, Fourth Edition

W. H. Inmon

WILEY
Wiley Publishing, Inc.

Building the Data Warehouse, Fourth Edition
Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN-13: 978-0-7645-9944-6
ISBN-10: 0-7645-9944-5

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

4B/SS/QZ/QV/IN

Credits

Executive Editor
Robert Elliott

Development Editor
Kevin Shafer

Production Editor
Pamela Hanley

Copy Editor
Kathi Duggan

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President & Executive Group Publisher
Richard Swadley

Vice President and Publisher
Joseph B. Wikert

Project Coordinator
Erin Smith

Graphics and Production Specialists
Jonelle Burns
Kelly Emkow
Carrie A. Foster
Joyce Haughey
Jennifer Heleine
Stephanie D. Jumper

Quality Control Technician
Leeann Harney

Proofreading and Indexing
TECHBOOKS Production Services

To Jeanne Friedman and Kevin Gould — friends for all times.

About the Author

Bill Inmon, the father of the data warehouse concept, has written 40 books on data management, data warehouse, design review, and management of data processing. Bill has had his books translated into Russian, German, French, Japanese, Portuguese, Chinese, Korean, and Dutch. Bill has published more than 250 articles in many trade journals. Bill founded and took public Prism Solutions. His latest company — Pine Cone Systems — builds software for the management of the data warehouse/data mart environment. Bill holds two software patents. Articles, white papers, presentations, and much more material can be found on his Web site, `www.billinmon.com`.

Contents

Preface for the Second Edition

Databases and database theory have been around for a long time. Early renditions of databases centered around a single database serving every purpose known to the information processing community—from transaction to batch processing to analytical processing. In most cases, the primary focus of the early database systems was operational—usually transactional—processing. In recent years, a more sophisticated notion of the database has emerged—one that serves operational needs and another that serves informational or analytical needs. To some extent, this more enlightened notion of the database is due to the advent of PCs, 4GL technology, and the empowerment of the end user.

The split of operational and informational databases occurs for many reasons:

- The data serving operational needs is physically different data from that serving informational or analytic needs.
- The supporting technology for operational processing is fundamentally different from the technology used to support informational or analytical needs.
- The user community for operational data is different from the one served by informational or analytical data.
- The processing characteristics for the operational environment and the informational environment are fundamentally different.

Because of these reasons (and many more), the modern way to build systems is to separate the operational from the informational or analytical processing and data.

This book is about the analytical [or the decision support systems (DSS)] environment and the structuring of data in that environment. The focus of the book is on what is termed the "data warehouse" (or "information warehouse"), which is at the heart of informational, DSS processing.

The discussions in this book are geared to the manager and the developer. Where appropriate, some level of discussion will be at the technical level. But, for the most part, the book is about issues and techniques. This book is meant to serve as a guideline for the designer and the developer.

When the first edition of Building the Data Warehouse was printed, the database theorists scoffed at the notion of the data warehouse. One theoretician stated that data warehousing set back the information technology industry 20 years. Another stated that the founder of data warehousing should not be allowed to speak in public. And yet another academic proclaimed that data warehousing was nothing new and that the world of academia had known about data warehousing all along although there were no books, no articles, no classes, no seminars, no conferences, no presentations, no references, no papers, and no use of the terms or concepts in existence in academia at that time.

When the second edition of the book appeared, the world was mad for anything of the Internet. In order to be successful it had to be "e" something—e-business, e-commerce, e-tailing, and so forth. One venture capitalist was known to say, "Why do we need a data warehouse when we have the Internet?"

But data warehousing has surpassed the database theoreticians who wanted to put all data in a single database. Data warehousing survived the dot.com disaster brought on by the short-sighted venture capitalists. In an age when technology in general is spurned by Wall Street and Main Street, data warehousing has never been more alive or stronger. There are conferences, seminars, books, articles, consulting, and the like. But mostly there are companies doing data warehousing, and making the discovery that, unlike the overhyped New Economy, the data warehouse actually delivers, even though Silicon Valley is still in a state of denial.

Preface for the Third Edition

The third edition of this book heralds a newer and even stronger day for data warehousing. Today data warehousing is not a theory but a fact of life. New technology is right around the corner to support some of the more exotic needs of a data warehouse. Corporations are running major pieces of their business on data warehouses. The cost of information has dropped dramatically because of data warehouses. Managers at long last have a viable solution to the ugliness of the legacy systems environment. For the first time, a corporate "memory" of historical information is available. Integration of data across the corporation is a real possibility, in most cases for the first time. Corporations

are learning how to go from data to information to competitive advantage. In short, data warehousing has unlocked a world of possibility.

One confusing aspect of data warehousing is that it is an architecture, not a technology. This frustrates the technician and the venture capitalist alike because these people want to buy something in a nice clean box. But data warehousing simply does not lend itself to being "boxed up." The difference between an architecture and a technology is like the difference between Santa Fe, New Mexico, and adobe bricks. If you drive the streets of Santa Fe you know you are there and nowhere else. Each home, each office building, each restaurant has a distinctive look that says "This is Santa Fe." The look and style that makes Santa Fe distinctive are the architecture. Now, that architecture is made up of such things as adobe bricks and exposed beams. There is a whole art to the making of adobe bricks and exposed beams. And it is certainly true that you could not have Santa Fe architecture without having adobe bricks and exposed beams. But adobe bricks and exposed beams by themselves do not make an architecture. They are independent technologies. For example, you have adobe bricks throughout the Southwest and the rest of the world that are not Santa Fe architecture.

Thus it is with architecture and technology, and with data warehousing and databases and other technology. There is the architecture, then there is the underlying technology, and they are two very different things. Unquestionably, there is a relationship between data warehousing and database technology, but they are most certainly not the same. Data warehousing requires the support of many different kinds of technology.

With the third edition of this book, we now know what works and what does not. When the first edition was written, there was some experience with developing and using warehouses, but truthfully, there was not the broad base of experience that exists today. For example, today we know with certainty the following:

- Data warehouses are built under a different development methodology than applications. Not keeping this in mind is a recipe for disaster.
- Data warehouses are fundamentally different from data marts. The two do not mix—they are like oil and water.
- Data warehouses deliver on their promise, unlike many overhyped technologies that simply faded away.
- Data warehouses attract huge amounts of data, to the point that entirely new approaches to the management of large amounts of data are required.

But perhaps the most intriguing thing that has been learned about data warehousing is that data warehouses form a foundation for many other forms of processing. The granular data found in the data warehouse can be reshaped and reused. If there is any immutable and profound truth about data warehouses, it is that data warehouses provide an ideal foundation for many other

forms of information processing. There are a whole host of reasons why this foundation is so important:

- There is a single version of the truth.
- Data can be reconciled if necessary.
- Data is immediately available for new, unknown uses.

And, finally, data warehousing has lowered the cost of information in the organization. With data warehousing, data is inexpensive to get to and fast to access.

Databases and database theory have been around for a long time. Early renditions of databases centered around a single database serving every purpose known to the information processing community—from transaction to batch processing to analytical processing. In most cases, the primary focus of the early database systems was operational—usually transactional—processing. In recent years, a more sophisticated notion of the database has emerged—one that serves operational needs and another that serves informational or analytical needs. To some extent, this more enlightened notion of the database is due to the advent of PCs, 4GL technology, and the empowerment of the end user. The split of operational and informational databases occurs for many reasons:

- The data serving operational needs is physically different data from that serving informational or analytic needs.
- The supporting technology for operational processing is fundamentally different from the technology used to support informational or analytical needs.
- The user community for operational data is different from the one served by informational or analytical data.
- The processing characteristics for the operational environment and the informational environment are fundamentally different.

For these reasons (and many more), the modern way to build systems is to separate the operational from the informational or analytical processing and data.

This book is about the analytical or the DSS environment and the structuring of data in that environment. The focus of the book is on what is termed the data warehouse (or information warehouse), which is at the heart of informational, DSS processing.

What is analytical, informational processing? It is processing that serves the needs of management in the decision-making process. Often known as DSS processing, analytical processing looks across broad vistas of data to detect trends. Instead of looking at one or two records of data (as is the case in operational processing), when the DSS analyst does analytical processing, many records are accessed.

It is rare for the DSS analyst to update data. In operational systems, data is constantly being updated at the individual record level. In analytical processing, records are constantly being accessed, and their contents are gathered for analysis, but little or no alteration of individual records occurs.

In analytical processing, the response time requirements are greatly relaxed compared to those of traditional operational processing. Analytical response time is measured from 30 minutes to 24 hours. Response times measured in this range for operational processing would be an unmitigated disaster.

The network that serves the analytical community is much smaller than the one that serves the operational community. Usually there are far fewer users of the analytical network than of the operational network.

Unlike the technology that serves the analytical environment, operational environment technology must concern itself with data and transaction locking, contention for data, deadlock, and so on.

There are, then, many major differences between the operational environment and the analytical environment. This book is about the analytical, DSS environment and addresses the following issues:

- Granularity of data
- Partitioning of data
- Meta data
- Lack of credibility of data
- Integration of DSS data
- The time basis of DSS data
- Identifying the source of DSS data-the system of record
- Migration and methodology

This book is for developers, managers, designers, data administrators, database administrators, and others who are building systems in a modern data processing environment. In addition, students of information processing will find this book useful. Where appropriate, some discussions will be more technical. But, for the most part, the book is about issues and techniques, and it is meant to serve as a guideline for the designer and the developer.

This book is the first in a series of books relating to data warehouse. The next book in the series is Using the Data Warehouse (Wiley, 1994). Using the Data Warehouse addresses the issues that arise once you have built the data warehouse. In addition, Using the Data Warehouse introduces the concept of a larger architecture and the notion of an operational data store (ODS). An operational data store is a similar architectural construct to the data warehouse, except the ODS applies only to operational systems, not informational systems. The third book in the series is Building the Operational Data Store (Wiley, 1999), which addresses the issues of what an ODS is and how an ODS is built.

The next book in the series is Corporate Information Factory, Third Edition (Wiley, 2002). This book addresses the larger framework of which the data warehouse is the center. In many regards the CIF book and the DW book are companions. The CIF book provides the larger picture and the DW book provides a more focused discussion. Another related book is Exploration Warehousing (Wiley, 2000). This book addresses a specialized kind of processing-pattern analysis using statistical techniques on data found in the data warehouse.

Building the Data Warehouse, however, is the cornerstone of all the related books. The data warehouse forms the foundation of all other forms of DSS processing.

There is perhaps no more eloquent testimony to the advances made by data warehousing and the corporate information factory than the References at the back of this book. When the first edition was published, there were no other books, no white papers, and only a handful of articles that could be referenced. In this third edition, there are many books, articles, and white papers that are mentioned. Indeed the references only start to explore some of the more important works.

Preface for the Fourth Edition

In the beginning was a theory of database that held that all data should be held in a common source. It was easy to see how this notion came about. Prior to database, there were master files. These master files resided on sequential media and were built for every application that came along. There was absolutely no integration among master files. Thus, the idea of integrating data into a single source — a database — held great appeal.

It was into this mindset that data warehouse was born. Data warehousing was an intellectual threat to those who subscribed to conventional database theory because data warehousing suggested that there ought to be different kinds of databases. And the thought that there should be different kinds of databases was not accepted by the database theoreticians.

Today, data warehousing has achieved the status of conventional wisdom. For a variety of reasons, data warehousing is just what you do. Recently a survey showed that corporate spending on data warehouse and business intelligence surpassed spending on transactional processing and OLTP, something unthinkable a few years back.

The day of data warehouse maturity has arrived.

It is appropriate, then, that the Fourth Edition of the book that began the data warehousing phenomenon has been written.

In addition to the time-honored concepts of data warehousing, this edition contains the data warehouse basics. But it also contains many topics current to today's information infrastructure.

Following are some of the more important new topics in this edition:

- Compliance (dealing with Sarbanes Oxley, HIPAA, Basel II, and more)
- Near line storage (extending the data warehouse to infinity)
- Multi dimensional database design
- Unstructured data
- End users (who they are and what their needs are)
- ODS and the data warehouse

In addition to having new topics, this edition reflects that larger architecture that surrounds a data warehouse.

Technology has grown up with data warehousing. In the early days of data warehousing, 50 GB to 100 GB of data was considered a large warehouse. Today, some data warehouses are in the petabyte range. Other technology includes advances made in multi-dimensional technology — in data marts and star joins. Yet other technology advances have occurred in the storage of data on storage media other than disk storage.

In short, technology advances have made possible the technological achievements of today. Without modern technology, there would be no data warehouse.

This book is for architects and system designers. The end user may find this book useful as an explanation of what data warehousing is all about. And managers and students will also find this book to be useful.

Acknowledgments

The following people have influenced—directly and indirectly—the material found in this book. The author is grateful for the long-term relationships that have been formed and for the experiences that have provided a basis for learning.

Guy Hildebrand, a partner like no other

Lynn Inmon, a wife and helpmate like no other

Ryan Sousa, a free thinker for the times

Jim Shank and Nick Johnson, without whom there would be nothing

Ron Powell and Shawn Rogers, friends and inspirations for all times

Joyce Norris Montanari, Intelligent Solutions, an inspiration throughout the ages

John Zachman, Zachman International, a friend and a world class architect

Dan Meers, BillInmon.com, a real visionary and a real friend

Cheryl Estep, independent consultant, who was there at the beginning

Claudia Imhoff, Intelligent Solutions

Jon Geiger, Intelligent Solutions

John Ladley, Meta Group

Bob Terdeman, EMC Corporation

Lowell Fryman, independent consultant

David Fender, SAS Japan

Jim Davis, SAS

Peter Grendel, SAP

Allen Houpt, CA

Building the Data Warehouse, Fourth Edition

CHAPTER

1

Evolution of Decision Support Systems

We are told that the hieroglyphics in Egypt are primarily the work of an accountant declaring how much grain is owed the Pharaoh. Some of the streets in Rome were laid out by civil engineers more than 2,000 years ago. Examination of bones found in archeological excavations in Chile shows that medicine — in, at least, a rudimentary form — was practiced as far back as 10,000 years ago. Other professions have roots that can be traced to antiquity. From this perspective, the profession and practice of information systems and processing are certainly immature, because they have existed only since the early 1960s.

Information processing shows this immaturity in many ways, such as its tendency to dwell on detail. There is the notion that if we get the details right, the end result will somehow take care of itself, and we will achieve success. It's like saying that if we know how to lay concrete, how to drill, and how to install nuts and bolts, we don't have to worry about the shape or the use of the bridge we are building. Such an attitude would drive a professionally mature civil engineer crazy. Getting all the details right does not necessarily equate success.

The data warehouse requires an architecture that begins by looking at the whole and then works down to the particulars. Certainly, details are important throughout the data warehouse. But details are important only when viewed in a broader context.

The story of the data warehouse begins with the evolution of information and decision support systems. This broad view of how it was that data warehousing evolved enables valuable insight.

The Evolution

The origins of data warehousing and *decision support systems (DSS)* processing hark back to the very early days of computers and information systems. It is interesting that DSS processing developed out of a long and complex evolution of information technology. Its evolution continues today.

Figure 1-1 shows the evolution of information processing from the early 1960s through 1980. In the early 1960s, the world of computation consisted of creating individual applications that were run using master files. The applications featured reports and programs, usually built in an early language such as Fortran or COBOL. Punched cards and paper tape were common. The master files of the day were housed on magnetic tape. The magnetic tapes were good for storing a large volume of data cheaply, but the drawback was that they had to be accessed sequentially. In a given pass of a magnetic tape file, where 100 percent of the records have to be accessed, typically only 5 percent or fewer of the records are actually needed. In addition, accessing an entire tape file may take as long as 20 to 30 minutes, depending on the data on the file and the processing that is done.

Around the mid-1960s, the growth of master files and magnetic tape exploded. And with that growth came huge amounts of redundant data. The proliferation of master files and redundant data presented some very insidious problems:

- The need to synchronize data upon update
- The complexity of maintaining programs
- The complexity of developing new programs
- The need for extensive amounts of hardware to support all the master files

In short order, the problems of master files — problems inherent to the medium itself — became stifling.

It is interesting to speculate what the world of information processing would look like if the only medium for storing data had been the magnetic tape. If there had never been anything to store bulk data on other than magnetic tape files, the world would have never had large, fast reservations systems, ATM systems, and the like. Indeed, the ability to store and manage data on new kinds of media opened up the way for a more powerful type of processing that brought the technician and the businessperson together as never before.

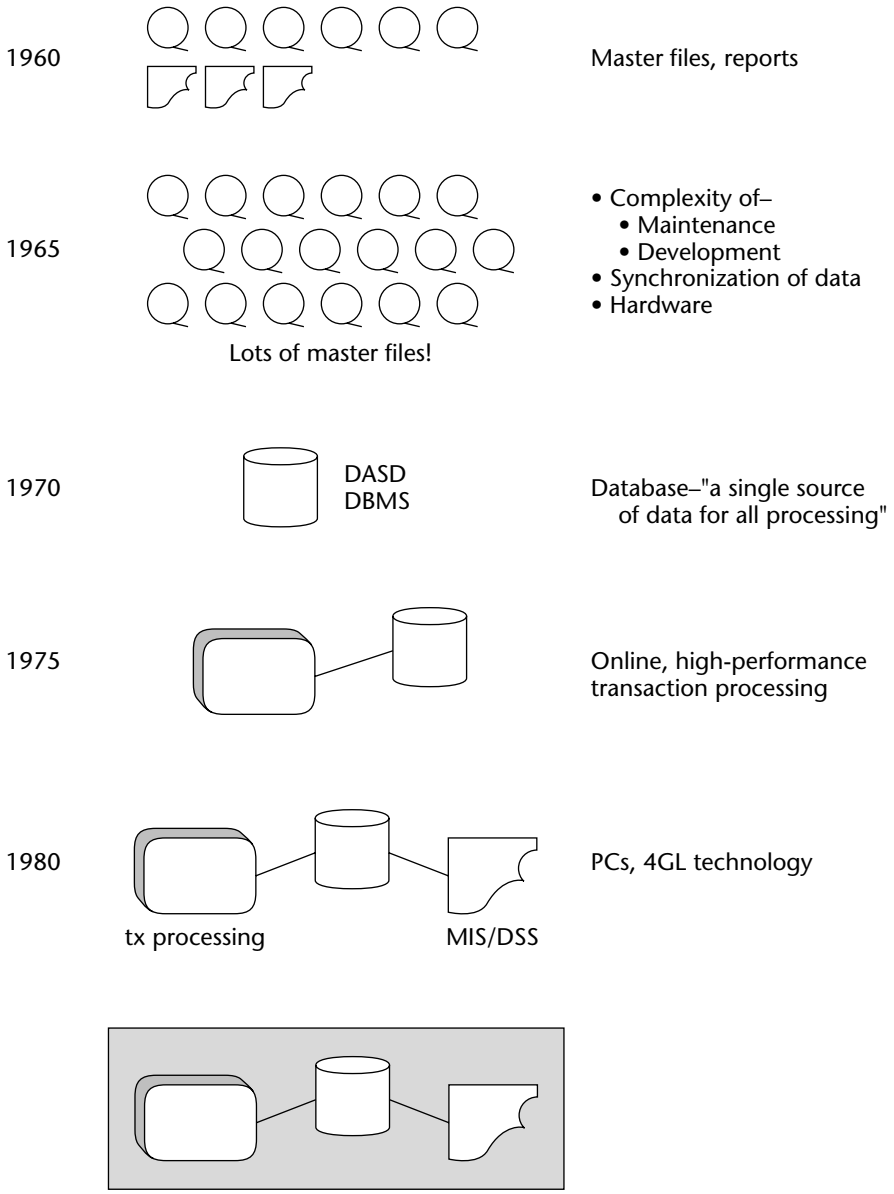

Figure 1-1 The early evolutionary stages of the architected environment.

The Advent of DASD

By 1970, the day of a new technology for the storage and access of data had dawned. The 1970s saw the advent of disk storage, or the *direct access storage device (DASD)*. Disk storage was fundamentally different from magnetic tape storage in that data could be accessed directly on a DASD. There was no need to go through records 1, 2, 3, . . . n to get to record $n + 1$. Once the address of record $n + 1$ was known, it was a simple matter to go to record n +1 directly. Furthermore, the time required to go to record $n + 1$ was significantly less than the time required to scan a tape. In fact, the time to locate a record on a DASD could be measured in milliseconds.

With the DASD came a new type of system software known as a *database management system (DBMS)*. The purpose of the DBMS was to make it easy for the programmer to store and access data on a DASD. In addition, the DBMS took care of such tasks as storing data on a DASD, indexing data, and so forth. With the DASD and DBMS came a technological solution to the problems of master files. And with the DBMS came the notion of a "database." In looking at the mess that was created by master files and the masses of redundant data aggregated on them, it is no wonder that in the 1970s, a database was defined as a single source of data for all processing.

By the mid-1970s, *online transaction processing (OLTP)* made even faster access to data possible, opening whole new vistas for business and processing. The computer could now be used for tasks not previously possible, including driving reservations systems, bank teller systems, manufacturing control systems, and the like. Had the world remained in a magnetic-tape-file state, most of the systems that we take for granted today would not have been possible.

PC/4GL Technology

By the 1980s, more new technologies, such as PCs and *fourth-generation languages (4GLs)*, began to surface. The end user began to assume a role previously unfathomed — directly controlling data and systems — a role previously reserved for the professional data processor. With PCs and 4GL technology came the notion that more could be done with data than simply processing online transactions. A *Management Information System (MIS)*, as it was called in the early days, could also be implemented. Today known as DSS, MIS was processing used to drive management decisions. Previously, data and technology were used exclusively to drive detailed operational decisions. No single database could serve both operational transaction processing and analytical processing at the same time. The single-database paradigm was previously shown in Figure 1-1.

Enter the Extract Program

Shortly after the advent of massive OLTP systems, an innocuous program for "extract" processing began to appear (see Figure 1-2).

The *extract program* is the simplest of all programs. It rummages through a file or database, uses some criteria for selecting data, and, on finding qualified data, transports the data to another file or database.

The extract program became very popular for at least two reasons:

- Because extract processing can move data out of the way of high-performance online processing, there is no conflict in terms of performance when the data needs to be analyzed en masse.
- When data is moved out of the operational, transaction-processing domain with an extract program, a shift in control of the data occurs. The end user then owns the data once he or she takes control of it. For these (and probably a host of other) reasons, extract processing was soon found everywhere.

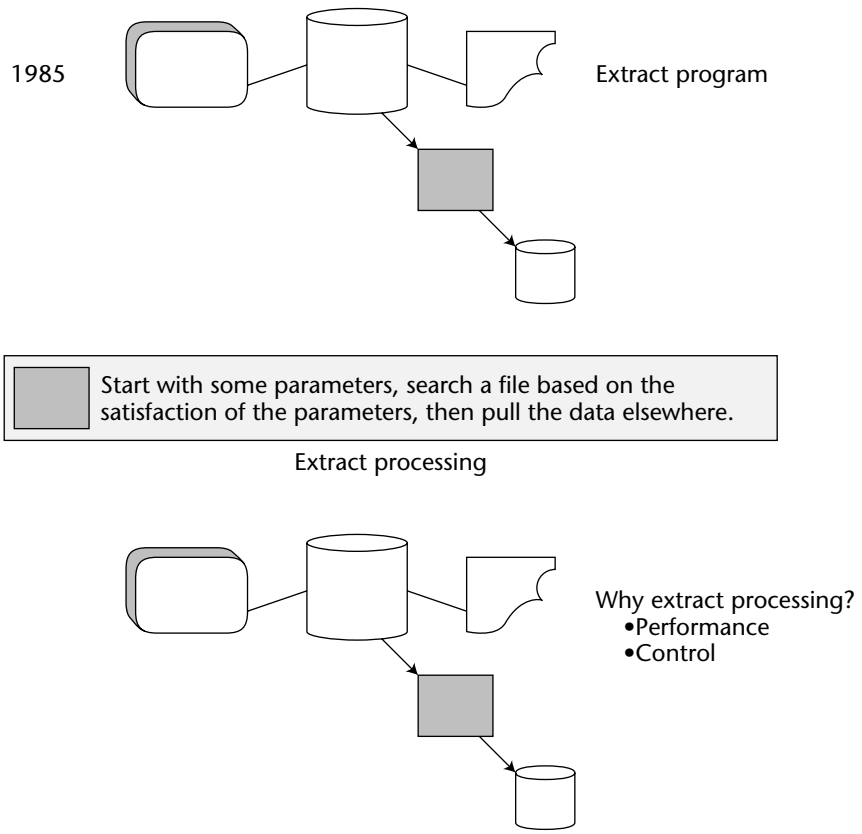

Figure 1-2 The nature of extract processing.

The Spider Web

As illustrated in Figure 1-3, a "spider web" of extract processing began to form. First, there were extracts; then there were extracts of extracts; then extracts of extracts of extracts; and so forth. It was not unusual for a large company to perform as many as 45,000 extracts per day.

This pattern of out-of-control extract processing across the organization became so commonplace that it was given its own name — the "naturally evolving architecture" — which occurs when an organization handles the whole process of hardware and software architecture with a laissez-faire attitude. The larger and more mature the organization, the worse the problems of the naturally evolving architecture become.

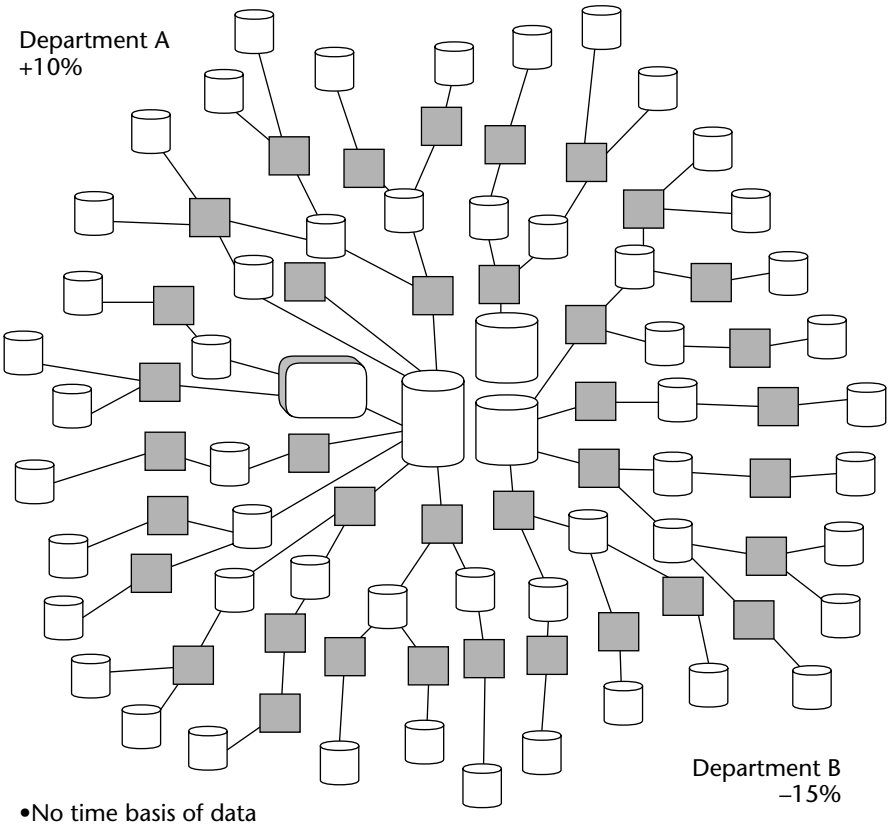

Figure 1-3 Lack of data credibility in the naturally evolving architecture.

Problems with the Naturally Evolving Architecture

The naturally evolving architecture presents many challenges, such as:

- Data credibility
- Productivity
- Inability to transform data into information

Lack of Data Credibility

The lack of data credibility was illustrated in Figure 1-3. Say two departments are delivering a report to management — one department claims that activity is down 15 percent, the other says that activity is up 10 percent. Not only are the two departments not in sync with each other, they are off by very large margins. In addition, trying to reconcile the different information from the different departments is difficult. Unless very careful documentation has been done, reconciliation is, for all practical purposes, impossible.

When management receives the conflicting reports, it is forced to make decisions based on politics and personalities because neither source is more or less credible. This is an example of the crisis of data credibility in the naturally evolving architecture.

This crisis is widespread and predictable. Why? As it was depicted in Figure 1-3, there are five reasons:

- No time basis of data
- The algorithmic differential of data
- The levels of extraction
- The problem of external data
- No common source of data from the beginning

The first reason for the predictability of the crisis is that there is no time basis for the data. Figure 1-4 shows such a time discrepancy. One department has extracted its data for analysis on a Sunday evening, and the other department extracted on a Wednesday afternoon. Is there any reason to believe that analysis done on one sample of data taken on one day will be the same as the analysis for a sample of data taken on another day? Of course not. Data is always changing within the corporation. Any correlation between analyzed sets of data that are taken at different points in time is only coincidental.

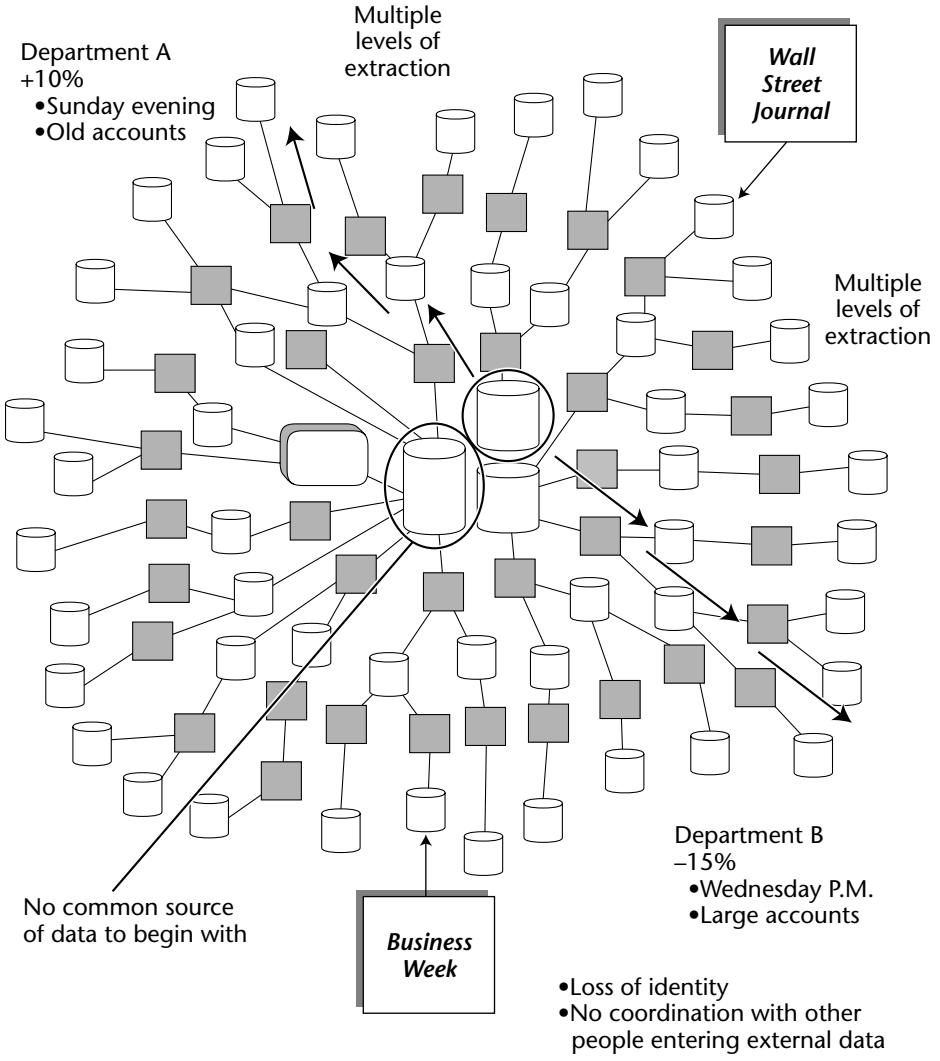

Figure 1-4 The reasons for the predictability of the crisis in data credibility in the naturally evolving architecture.

The second reason is the algorithmic differential. For example, one department has chosen to analyze all old accounts. Another department has chosen to analyze all large accounts. Is there any necessary correlation between the characteristics of customers who have old accounts and customers who have large accounts? Probably not. So why should a very different result surprise anyone?

The third reason is one that merely magnifies the first two reasons. Every time a new extraction is done, the probabilities of a discrepancy arise because of