Chapter 1

A Crash (or Collision) Course: A History of Data Warehousing

He who does not remember history is condemned to repeat it.
—George Santayana

The Current State of Data Warehousing and E-Commerce

The focus of individual and collective consciousness naturally tends to be drawn to the new and different, to the disadvantage of the old and familiar. It's not surprising, then, that most of the attention surrounding e-commerce is focused on the enabling hardware and software. In e-commerce, the world-wide network infrastructure, the Web browsers and Web servers, are the fundamentally newer components, therefore garnering the most continuous attention. However, without data, the increased sharing of which is, after all, the underlying basis for the Information Age, all this enabling infrastructure is but an empty frame.

This book will focus on the data aspects of e-commerce. There are of course countless other aspects of e-commerce, but in our admittedly simplistic, data-centric view of the digital economy, everything else that happens in e-commerce is just stuff that happens to data.

When it comes to the data, like *Alice in Wonderland's* Queen of Hearts, sometimes it seems as if we need to run as fast as we can to stay in one place. Here in the early 21st century, we're still trying to get our hands around all

the data we accumulated in the late 20th century. As soon as a glimpse of light appears at the end of the tunnel, another IT revolution—client/server, the Web, and who knows what's next—confounds our best efforts, and it seems we are nearly back at square one. All the while, more data continues to accumulate, faster and faster, and in increasingly far-flung locations and disparate formats.

How best should we approach this dilemma, in order to assure our greatest chance of future success? By building on lessons learned from the successes of the past.

If sustained growth is an accurate indicator, data warehousing is by far one of the most successful information technology strategies in the history of commercial computing. For example, in a report by technology research firm IDC, worldwide revenue in the market for data warehouse tools is projected to "increase at a compound annual growth rate (CAGR) of 26 percent from $5 billion in 1999 to $17 billion in 2004."

As a first step in learning how to leverage our past successes for tomorrow's markets, let's take a look at the current state of data warehousing and e-commerce technologies, and how we got to where we are today. To get our bearings, here are some widely accepted definitions for both data warehousing and e-commerce. The most authoritative names in data warehousing define *data warehouse* as

> A collection of integrated, subject-oriented databases designed to support the DSS function, where each unit of data is specific to some moment of time. The data warehouse contains atomic data and lightly summarized data.
>
> —Bill Inmon, *Building the Data Warehouse*

> A copy of transaction data specifically structured for query and analysis.
>
> —Ralph Kimball, *The Data Warehouse Toolkit*

And from a similarly authoritative source on e-commerce:

> Electronic commerce is the buying and selling of goods and services, and the transfer of funds, through digital communications.
>
> —Ted Haynes, *The Electronic Commerce Dictionary*

In this book, we'll be focusing much more on *data warehousing* than on *data warehouses* per se. Data warehousing, in our context, refers to the process and the technologies (the "hows") used to build, maintain, and use data warehouses (the "whats"). Therefore, our objective is to learn *how to*

leverage our prior successes in the process and the technologies used to build, maintain, and use data warehouses to enhance our chances of success in the buying and selling of information, products, and services via computer networks.

The market for data warehouses and data marts as autonomous, bundled sets of technologies has evolved and matured. No longer limited to early adopters, they have been deployed by many companies across the technology-adoption curve. Regardless of the success or failure of such efforts, the goals of data management architecture are expanding to encompass more than simply the construction of consolidated data sources. These goals now include the broader mission of making the greatest possible use of any and all available data assets.

Data requirements in today's electronic marketplace are challenging and will become even more so. E-commerce sites have been broadly categorized into business-to-consumer (B2C) and business-to-business (B2B). Many B2C Web sites are of course sales channels for retail businesses, where customers purchase goods sold by the business. Other common B2C sites are for financial services such as banking. These two types vary in their data content, and therefore data-integration requirements.

For example, the data content of a financial services site might include

- *Bank Accounts:* checking, credit, loans, mutual funds, brokerage
- *Products:* interest rates, terms, fees
- *Assets:* holdings in the accounts
- *Transactions:* bill payments, inquiries, account transfers

Financial companies also typically have multiple application systems (usually one per product line) that are the source or target, or both, of the data presented by the Web site's pages.

In contrast, the content of a retail business-to-consumer Web site typically includes the following types of data:

- *Product Catalog Items:* what is being sold
- *Inventory:* how much of each product is available (currently and projected)
- *Order:* how much of what has been ordered by whom, at what price
- *Credit:* how payment is to be transferred between buyer and seller
- *Shipping:* how the products are to be transferred between seller and buyer

- *Customer Profile:* everything the seller has been able to find out about the buyer

Often, these discrete types of data are stored and processed by a separate application system. Complicating matters further, only some—and potentially none—of these systems may be owned and operated by the retailer itself! Establishing and maintaining a reliable infrastructure enabling data to freely and reliably cross organizational and technological boundaries among multiple businesses often requires an unprecedented level of cooperation and coordination.

Existing heritage databases will not be replaced, but must be wired in and synchronized with the databases of multiple inbound and outbound channel applications. (More details on channels will be discussed in Chapter 2.) Data originating through inbound and customer-facing channels must be synchronized with back-office operational data. Orders coming in over the Web must be matched to inventory records, perhaps back-ordered with suppliers, then scheduled for delivery.

A prime example of this is amazon.com. This Web site enables a customer to not only establish an account and immediately begin placing orders, but also to track the status, including shipping information, of any order. A new Web-based business enterprise such as this has the luxury of near freedom from any stovepipe order-entry, inventory, and shipping applications. Such a business, *created* for the digital economy, may be able to enter the market armed and ready with an integrated value chain—and a cohesive data resource—by design. But on the downside, few of these new digital enterprises provide choices of channels beyond the internet…yet.

Such customer-facing Web sites continue to dramatically and irrevocably alter the expectations of their users. The external constituents (and not just customers, as we shall see) of an aspiring digital enterprise will not be content in an environment where the physical location or format of stored data is visible or of any consequence whatsoever. There is no doubt, however, that removing these barriers remains a major challenge confronted by digital business.

On the B2B side, the data challenges are at least as daunting as in B2C. Whereas B2C applications are characterized by large numbers of individual, small events taking place between humans and computers, a typical B2B application deals with fewer participants, and with events that involve larger dollar amounts, taking place in the context of computer-to-computer as well as human-to-computer interactions. In B2C e-commerce, the seller as a rule has control of the format of the data being exchanged. In B2B, the buyers, fewer in number but each with considerably more individual leverage over the seller than in B2C, may in many cases exercise as much or more control over

the format of the data being exchanged than does the seller. This state of affairs poses additional data-integration challenges for the seller and presents an opportunity for B2B exchanges, which we will explore in more detail later.

It's the Data, Stupid! The Means to Many Ends

Data is, and will continue to be, the most stable, most slowly-evolving component of any company's information technology assets. It is the component most independent of the technology itself. It is also unique to each business enterprise—a significant source of its competitive differentiation—and, as such, an invaluable resource. Data is accumulating at accelerating rates, faster than software and hardware technology can address it; hence, hardware and software bottlenecks continue to constrain its usefulness. Not only is a history of facts being accumulated with the passage of time, but more and more replicated information is being created as a result of hardware and software bottlenecks. Software is required both to create the data records and to interpret the accumulated information, causing a need for more software and interfaces.

When computers were first applied to business use, the field of endeavor was named *data processing.* Now, of course, we know the field as *information technology.* But, as a way of getting back to basics, let's look at this thought problem.

A business has a choice of one, and only one, of the following disaster plans:

1. In the event of a disaster, all *hardware devices* and *software programs* in the company's IT inventory—*but no data*—are instantly duplicated at an alternate site, or
2. In the event of a disaster, all of the company's *data only* is instantly copied to an alternate site.

Which is the most prudent choice? Which will enable the business to continue functioning *at all?* Does it then follow that it makes sense to focus our priorities on the unique data resources of the business?

Another way of approaching the problem is from the view that a company's competitors could probably duplicate exactly its entire hardware inventory and the majority of its software inventory, *but its unique data cannot be duplicated* (at least not legally)!

The history of data warehousing is in many respects the story of the ongoing effort to get more value out of the data we have already accumulated, or, stated another way, to put large volumes of data resources to work. The goal is to transform data into action. This history is also characterized by *convergence*. As Figure 1–1 shows, over time many concepts and technologies have been introduced, and consequently influenced and converged with other concepts and technologies. In some cases, ideas may have been introduced before enabling technologies had matured, and when maturity did come, the enabling technology acted as a catalyst.

It's the contention of this book that with the convergence of data warehousing and e-commerce, data warehousing technologies have begun to act as catalysts for e-commerce—and that this catalysis will only intensify as the digital economy matures.

Figure 1–1 describes the progression of various trends in data warehousing technologies over the past 15 years, and how these trends have influenced, morphed into, merged into, or instigated other developments. We'll be discussing and interrelating many of these trends throughout this book.

Business Trends and the Use of Data

Electronic commerce predates the Internet and the explosion of the World Wide Web. Pre-Web e-commerce applications such as EDI (Electronic Data Interchange) were computer-to-computer data exchange; they were *purely* data-intensive. ("Have my computer call your computer.") Furthermore, these data-interchange applications were extremely structured—file layouts, semantics, and syntax were preestablished and rigidly controlled by standards organizations, suppliers, and service providers. Connections were one-to-one. Applications were stable for years at a time. Transfers of data usually took place only a few times daily, after business hours, not on weekends, and seldom crossed international borders.

The Web has brought e-commerce to the masses. Early commercial Web sites were far from data-intensive, being primarily *brochureware*—an electronic billboard, another channel for advertising. The databases were connected to Web sites in read-only mode.

As can be seen in Figure 1–1, from our data-oriented perspective, e-commerce is the latest in a long line of business initiatives that require a broad scope of data—a scope beyond that of a single application system. The most

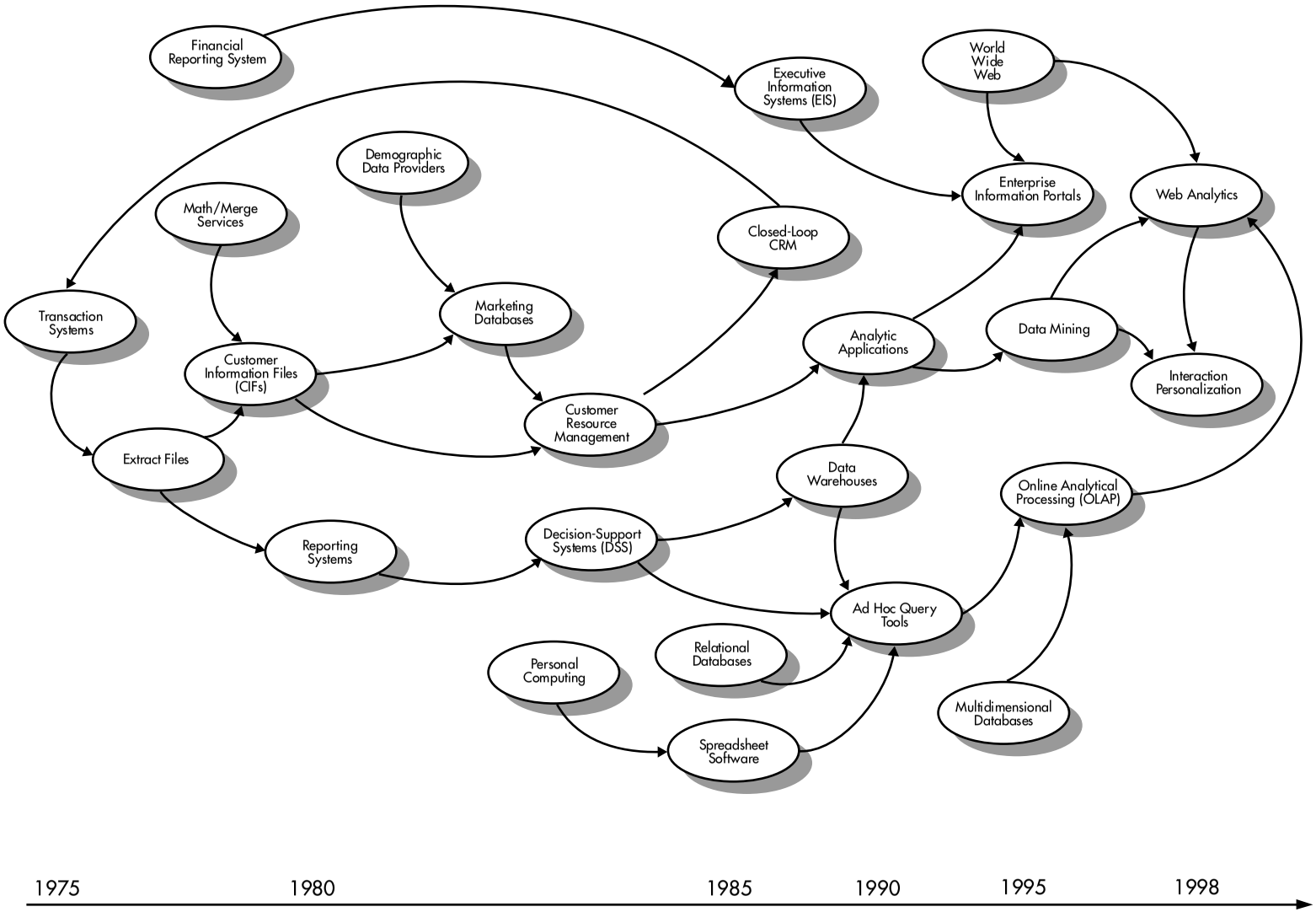

Figure 1–1
Trends and influences in data warehousing, 1975–2000.

common application of data warehousing processes and technologies in business are in support of these initiatives, many of which, as we shall see, are morphing into new forms integral to the world of e-commerce.

The majority of business application systems are built—*applied*—in support of specific business functions. Decision-making is of course a business function, but, in contrast to most other business functions, it creates little data on its own; it is a net *consumer* of data, rather than a *producer*. In the family of business applications, it's the parasitic brother-in-law who just won't leave. It commandeers the television, lays on the couch, raids the refrigerator, and consumes whatever resources it finds available. Luckily, as opposed to "real" resources like food and clothing, in the case of less tangible data resources, we do indeed have a replicator (remember the one in *Star Trek*?) available. We can create copies of data (as many as we need or want) from the applications that do the "real" work for the exclusive use of our decision-making needs.

Thirty or so years ago, under the cover of darkness, reporting systems began to leach data from transactional applications' files and create their own extracts, in formats more easily consumable by corporate decision makers. Impact printers banged out stacks of hardcopy "batch" reports on nightly, monthly, quarterly, and annual bases. Tape reels containing the extracted data were organized, labeled, and hung up in case a report was lost and needed to be rerun.

We'll now take a look in more detail at some of these trends in data-intensive business initiatives. Most of these initiatives came about due to the critical needs of companies to know and understand their finances, their customers, and "the big picture" of the company as a whole.

Know Your Finances

In many companies, the first department that made use of computer application systems was finance. And, predictably, the first business unit requesting reports and extracts for decision-making was the finance department.

Financial Reporting Systems

General ledger (GL) systems were the original decision-support systems: net consumers of data, existing almost exclusively to generate reports. GL systems create little data of their own and require a holistic view of all the financial comings and goings of a business. In many ways, a GL system does look much like a decision support system (DSS), but since the auditability of GL data is absolutely critical, support for alternative views—such as "what-if" analysis

and cost and revenue projections—required construction of decision-support capabilities separate and isolated from the accounting system itself. Often, these included copies of the general ledger account files, perhaps organized or selected according to specified criteria (by product or organization, for example) and accessible to users without jeopardizing the source data itself.

Profitability Measurement: Customer, Product, Organization, and Channel

Profitability measurement also requires an enterprise-wide perspective on an organization's data assets. Profitability in general is the measurement of revenue relative to costs within a specific time period and a defined scope—commonly for individual or groups (segments) of customers, products, business units, and/or sales channels. Because of these multiple "dimensions" by which cost and revenue data needs to be viewed, this area has been a fertile one for the application of multidimensional databases and online analytical processing (OLAP) tools.

Continuous monitoring of costs, revenues, and profitability is required for a digital enterprise to remain responsive to changing market and competitive conditions. Monitoring and analysis of transaction activity for customers and products has been common in businesses for some time, but the evolution toward a multichannel interaction model by the digital enterprise mandates that costs, revenues, and profitability *across channels* also be monitored and analyzed. The goal is to both increase the profitability of the most active channels and concurrently increase the activity of the most profitable channels.

Many inbound channels capture data on literally every telephone button-push or mouse-click in every interaction of every customer. This data can be analyzed to determine, for example, how often customers decide to "zero out" to a human operator, rather than using menu options to finish their transaction. A high volume of zeroing-out indicates that menu options and structure may require changes, and also that the per-transaction costs have escalated due to increased involvement of customer service representatives (CSRs). CSRs require salaries, benefits, real estate, and safe working conditions; computer-based interaction units are much less expensive. If electronic channels are made more effective, human interactions are decreased, costs are driven down, and profits rise.

Activity-Based Costing

Activity-based costing (ABC) is closely related to profitability measurement, since capturing, allocating, and measuring costs is essential to calculating

profitability. Numerous data-management challenges arise when attempting to define and track costs of activities within the organization. In order to be meaningful, a consistent activity cost measurement structure must be put in place across an entire company. Challenges include achieving all of the following, consistently, throughout the company:

- establishing the scope of what constitutes an "activity"
- capturing activities
- classifying and allocating activities against the production of revenue

If a company can achieve consistent data and process standards for cost accounting, they are far along the way to implementing ABC, and subsequently more meaningful profitability measurement.

Know Your Customer

Whereas transactional systems deal primarily with keeping the business running on a *daily* basis, a class of decision support systems known as *customer management systems* (CMSs) are concerned with optimizing the decision-making of sales and marketing organizations by taking a *long-term* view of the interactions between the business and its customers. The evolution of CMSs has primarily been an effort to steadily decrease the lag time between opportunities for outbound interactions with customers and prospects—from a few times annually to literally real-time.

First-generation marketing systems (mid-1970s to early 1980s) primarily supported intermittent marketing programs, such as direct-mail and mass-marketing campaigns. These programs were executed on a monthly (or less frequent) basis and as such were supported quite adequately with sequential extract files and batch processes. Early on in this generation, companies such as Harte-Hanks and Claritas appeared, from whom demographic data could be purchased for merging with and "enriching" a company's internal transaction data. The resulting increased awareness of geographic, affluence, and household data allowed marketing campaigns to be targeted to those prospects and customers having a potentially greater buying propensity, hence increasing the success rate and ROI of campaign efforts.

In the later 1980s, as marketing DSS evolved into a second generation, more emphasis was placed on increased and more frequent accessibility to extracted and consolidated data, often facilitated by the creation and population of a stand-alone relational database. At the same time, to remain compet-

itive, many organizations attempted to change their fundamental strategy from "product-centric" to "customer-centric." Consolidating data into a single source from multiple transaction systems and attempting to derive customer data from transaction data exacerbated an already existing problem. For multiple sales transactions, the same customer information had often been solicited and recorded multiple times. Companies thus found themselves with "too much" customer information—often with duplicate or nearly duplicate records for the same customer—one for John Smith, another for John A. Smith, another for Jack Smith, and so forth.

An early attempt at solving this problem was the customer information file (CIF) concept. A CIF is closely related to a marketing database. The goal of building a CIF is usually to extract and de-duplicate ("de-dupe") multiple customer records, consolidating multiple redundant records into one, and also to relate to the consolidated customer record the *products* sold to each given customer. This CIF-building process was widely implemented in financial services organizations. Concurrently, other companies emerged that specialized in software and services for matching and merging customer records, as well as deriving "household" relationships from that data, based on matching last names, addresses, telephone numbers, and so forth.

The current, third generation of customer data management is widely referred to as customer relationship management (CRM). This generation has been strongly influenced by the increased control assumed by the customer made possible by the proliferation of delivery channels. This in turn has driven needs for "mass customization" techniques such as personalization and one-to-one marketing. Also coming into play in CRM is the requirement to "close the loop" between *front-office* (customer-facing) processes and *back-office* (internal operations) processes. In a closed-loop scenario, the results of analytical interactions are "fed" back into the information value chain at one or more points, either into the DSS system itself, or in the most advanced configurations, back into the operational systems. Figure 1–2 illustrates an example of a closed-loop scenario.

The World Wide Web affords businesses an additional "touch-point" for inbound and outbound interactions with customers and prospects. The extension of a company's presence to the Web offers the opportunity for the company to not only know their customers better, but also to capture information, from sparse to rich, on the interactions of *anyone* who "hits" their Web site. The sparsest of information is available on "anonymous" browsers; incrementally, more data can be captured as a function of the intimacy of the browser's interaction with the site—accepting cookies, registering for a logon ID, and through execution of transactions. More on this topic—Web analytics—is covered in Chapter 4.

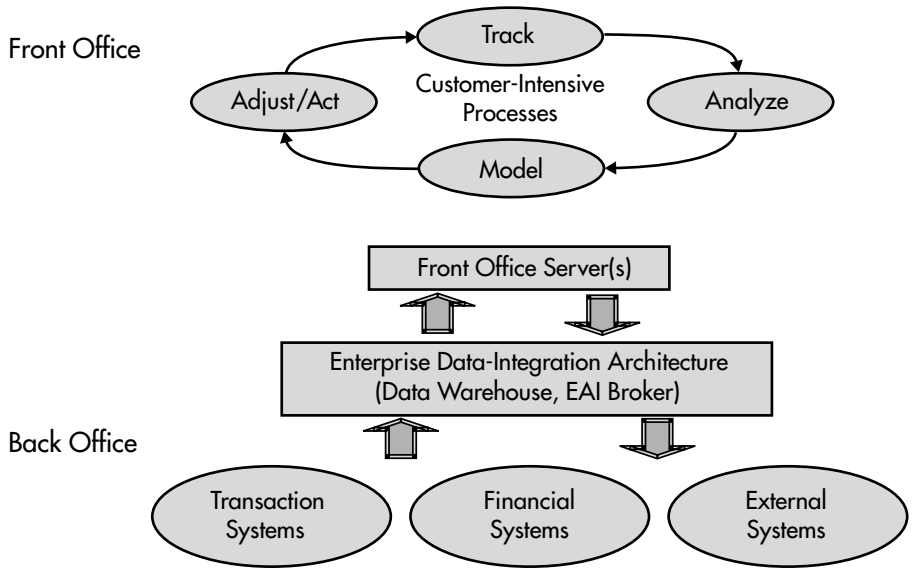

Figure 1–2
Closed-loop CRM.

Know Your Company

Truly "knowing the company" requires management to keep an eye on more than just the day-to-day financial details. The functioning of the company needs to be seen in a contextual perspective—requiring dynamic comparisons between current and past performance, projecting trends into the future, and benchmarking corporate metrics against the competition. Several techniques have been originated over the past decades to assist executives in keeping a finger on the corporate pulse.

Executive Information Systems (EIS)

The concept of the executive information system is an example of a wine being sold before its time. In the late 1980s, several vendors began offering what they dubbed executive information systems. EISs, essentially an output and data-visualization metaphor, purported to be an easy-to-read "dashboard" by which busy (and marginally computer-literate, to be honest) executives could monitor the overall "health" of their organization. EISs were positioned as

decision-support systems that made decision-making information available "at a glance."

Most EIS implementations were dismal failures, precisely because the colossal behind-the-scenes data acquisition and integration issues were purposely ignored or downplayed by vendors. To be of any use, an EIS required data of enterprise scope. But at the time they were introduced, there were as yet no commercially available data-consolidation products, and the very few consolidation efforts that had been undertaken were confined to supporting decision-making of a more limited scope. When inquiries were made to internal information systems (IS) shops regarding the feasibility of acquiring and consolidating the data necessary to really "do" EIS, the response was overwhelmingly incredulous. There was no way for this integration effort to be accomplished within the time and cost ballpark represented to unsuspecting executive managers by many EIS vendors. IS was placed in a no-win situation.

However, the overall concept of the EIS as a 20,000-foot view of the entire company, after laying dormant for about a decade, was given new life by the ubiquity of Web browsers and data-consolidation architectures—and the result is now called Enterprise Information Portals, or EIPs.

Balanced Scorecard

The balanced scorecard concept was described by Robert S. Kaplan and David P. Norton in their book *The Balanced Scorecard: Translating Strategy into Action* (Harvard Business School Press, 1996). The scope of its data requirements is probably the broadest of all the DSS approaches discussed, encompassing data on financials, customers, costs, and more. It specifies four perspectives to be monitored on an ongoing basis by executive management:

- *Financial:* company financial resources
- *Customer:* how customers see the organization
- *Internal:* internal processes and activities
- *Learning and Growth:* continuous improvement capabilities

Balanced scorecard concepts complement EISs nicely as a pattern for classifying and presenting information within an EIS. Commercial applications have indeed built around this combination, including offerings from CorVu Corporation and Gentia Software.

Enterprise Information Portals

Enterprise Information Portals entered the scene in the late 1990s. The concept of an EIP is essentially that of an EIS for the masses. With an EIP, widespread employee access to a broad range of corporate data is enabled through the Web browser software on their desktops. However, EIPs, being front-ends like EISs, require the same back-end, data-consolidation plumbing as an EIS. Data warehousing and related data-management technologies, developed and widely implemented since the introduction of the EIS concept, have made EIPs more feasible than the EIS predecessors.

Now we've seen a number of common business applications that require access to a broad base of corporate data. In the next section, we'll dig deeper into the technologies that enable these "data warehousing-like" applications, and which can also give companies a running start into the e-commerce mêlée.

Technological Underpinnings

Throughout this book, we will discuss the technological underpinnings of data warehousing and e-commerce in terms of three broad areas, as follows:

1. *Output:* including analytic applications, business intelligence, user-data interaction models
2. *Storage:* software and hardware that is specialized for storage of data—including database management systems and high-performance servers
3. *Input:* including data movement, replication and transformation

In the remainder of this historical overview chapter, we will work our way from front to back—front office to back office—and briefly review the evolution of technologies in these three areas. In subsequent chapters, we'll discuss in detail how these areas fit into a company's e-commerce strategy. But first, a clarification of some terms may be in order.

Data Warehouses and Data Marts, Briefly

Bill Inmon is widely credited with popularizing the data warehouse concept and terminology; in fact, his Web site *(www.billinmon.com)* christens him

the Father of Data Warehousing. Objectively, his 1992 book, *Building the Data Warehouse*, as well as numerous articles and presentations before and since, must be credited with propagating a common language around decision-support technologies—which in turn formed a catalyst for a great deal of successful software technology and human effort.

One of Inmon's definitions of a data warehouse was presented early in this chapter. Another, probably more widely quoted definition he has also offered is

> A (data) warehouse is a subject-oriented, integrated, time-variant and nonvolatile collection of data in support of management's decision-making process.
>
> —Bill Inmon

By way of additional explanation, Inmon provides these definitions:

- *Subject-oriented:* Data that gives information about a particular subject instead of about a company's ongoing operations.
- *Integrated:* Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole.
- *Time-variant:* All data in the data warehouse is identified with a particular time period.
- *Nonvolatile:* Data is stable in a data warehouse. More data is added, but data is never [never?] removed. This enables management to gain a consistent picture of the business.

Much has been written and spoken by Inmon, Ralph Kimball, and others comparing and contrasting *data warehouses* and *data marts*. For most intents and purposes, a data mart is a little data warehouse. Bill Inmon, Ralph Kimball, and Alan Simon are well-respected authorities in data warehousing and have written and spoken widely on the subject. Information on their recommended writings is available in the Bibliography.

Since the purpose of this book is not to investigate data warehouses themselves, we will not go into much more detail on these topics here. An abundance of books is available on planning, modeling, designing, building, managing, tuning, using, and just about any other thing anyone would possibly wish to do to or with a data warehouse. That said, however, one point fundamental to the premise of this book needs to be made here:

In any given e-commerce effort—regardless of whether or not the data within the scope of the effort meets any or all of the commonly stated criteria for a data

warehouse or data mart—the tools, techniques, and technologies commonly used to create, manage, and use data warehouses and data marts should be given serious consideration.

Output: From Reporting to Portals

All the behind-the-scenes data warehousing processing and technology—input and the storage—are means to an end: the output. The output is all about enabling more informed decision-making by the employees of an organization, and increasingly, for its customers as well.

When discussing output technologies, we're speaking about the human interface—the interactions between data and consumers of the data. A primary characteristic of the evolution of the DSS human interface has been the degree of *interactivity* afforded the end user by data-access software. Over the past three decades, data output technologies have changed dramatically. Throughout the 1970s, hardcopy reports were the norm; interactivity was limited to the visual, and perhaps extended to writing notes or highlighting on the reports. Today users have available highly–interactive, online, analytic processing and visualization tools, where selected data can be formatted, graphed, drilled, sliced, diced, mined, annotated, enhanced, exported, and distributed. The current culmination of this evolutionary path is the concept of *closed loop* business intelligence.

With early batch reporting systems, what the user saw—a specific set of "canned" reports—was what the user got. If any changes or additions to the reports were needed, a change request needed to be submitted and queued up with the IT shop. And since these systems were all customized, in-house creations, and the IT shop had the software firmly under its control, the turnaround rate for changes and enhancements was often slower than the user would have liked (an understatement, perhaps). An example of the type of technology available during this period is the programming language RPG (Report Program Generator), introduced by IBM in 1965. RPG was designed specifically to enable rapid development of batch reports.

During the late 1970s and early 1980s, both in-house development shops and newly established software vendors seized the opportunity to provide users with a greater degree of direct interactivity with their data. Software evolved that accepted input, in the form of parameters, by which the end user could select specific desired data and also specify the format in which the output appeared. No longer were separate reports needed for each sort sequence, selection criteria, field subset, and so on; the users were now able to specify what they wanted, when they wanted it.

Users' desires for this type of interactivity gave rise to a robust software industry that continues to this day. Trailblazers in the reporting software market were FOCUS (Information Builders, Inc.), Ramis (Computer Associates International, Inc.), and SAS (SAS Institute, Inc.). Each of these products originated during this period; each has grown and is going strong today, in its seventh or eighth major release.

Generally, in order for this type of software tool to function, any data to be analyzed needed to be copied and transformed into the specific format required by the tool, for example, into FOCUS databases or SAS files. This constraint actually pointed these vendors in the directions we know today as database middleware and extract-transform-load (ETL) tools.

The widespread availability of personal computers beginning in the mid-1980s presaged a significant change in the delivery and presentation of decision-making information. The most powerful change agents in PC software included, most significantly on the output side, spreadsheet programs such as VisiCalc, Lotus 1-2-3, and eventually, Microsoft Excel. These spreadsheet products gave users their first taste of "interacting with data."

The PC revolution, in putting graphical horsepower on knowledge workers' desks, moved end-user reporting tools such as SAS and FOCUS onto the desktop, as well as triggered the explosive growth of desktop reporting tools such as Brio.Enterprise, Cognos Impromptu, and Business Objects.

In the mid-1990s, users' ability to interact with data on their desktops took a huge leap forward, with the emergence of OLAP software products. OLAP enables a view of data based on the spreadsheet metaphor. Using an OLAP tool, a user can pretty much instantaneously alter the sorting, selection, intersection, and hierarchical arrangement of the data being observed. As we will see, OLAP functionality is critical in enabling access to the large amounts of data being generated through e-commerce interactions. In fact, most reporting-tool vendors have since expanded their offerings to support OLAP functionality.

Data mining software has also seen phenomenal growth in the last decade. Descended from artificial intelligence research and statistical modeling, data mining functions use large amounts of data to draw conclusions. Whereas query and even OLAP functions require human interaction to follow relationships through a data source, data mining programs are able to derive many of these relationships automatically by analyzing and "learning" from the data values contained in files and databases.

The concept of EIPs arose as a result of the Web-enabling of decision-support systems. Until the advent of the ubiquitous Web browser, in order to interface with a data warehouse, users typically needed to either have query software installed and configured on their workstation, or at the very least, had

to be on the distribution list for—guess what—hardcopy printouts. This led inevitably to challenges in software cost, distribution, and maintenance, and to "killing more trees"—the elusive paperless office remained so. But with, in effect, a browser on every desktop in the workplace, Web-enabled query software extended the visibility of the data warehouse to potentially every authorized knowledge worker in an organization—and beyond.

Storage: Extract Files to Objects

In the 1960s, thanks to Herman Hollerith, we first codified the bulk of our business data in a digitally processable format: punched cards. Throughout the 1970s, precursors to "true" decision-support systems were constrained to reporting from punched cards or from the newer sequential media, tape reels. In the 1980s, we transferred most of our most dynamic data assets to the new, random-access disk drives, leaving the history on magnetic tape. Random access was a revolutionary development. In the last three decades of the 20th century, remarkable progress was made in data storage technologies. Now, as we begin a new millennium, we are reading data from smaller, faster, and cheaper…disk drives. When is the next revolution? Disks are being made faster and faster, more data is being packed into less space, but eventually, pure physics takes over. Only so much can be done with the current fundamental storage medium, iron oxide—rust. More on data storage media will be covered in Chapter 5.

The primary enabler for access to large amounts of data was what could be termed "the random-access revolution." Just as it is much easier and faster to select a particular song from a CD (or vinyl LP) than from a cassette tape, it's much easier and faster for a decision maker to select specific data from a revolving disk than from a tape. A disk is always "mounted" in place, whereas, even given advances in automated robotic tape silos, a tape must be located, taken to a reader, and mounted on the reader. Then, every record on the tape must be read up to the point where the required data is located—even if it is at the end. These days, of course, we take random-access disk storage for granted; we have gigabytes even on our laptops and home PCs. But back in the 20th century, random-access storage media fundamentally changed the relationship between data and its users. The most significant result of this was the rise of software systems specialized for managing the storage and retrieval of large, structured sets of data—databases. These products were dubbed database management systems, or DBMSs.

In the mid-1980s, decision-support data storage was dealt a double

whammy: the almost simultaneous rise of personal computing and the productization of the relational database model. At nearly the same time that relational databases became available on mainframe and midrange computers, "personal" databases—relational and relational-like, such as Borland's Paradox and Ashton-Tate's dBase—began to grace the desktop PCs in homes and offices.

Relational databases achieved a level of user-friendliness unheard of in earlier hierarchical and network database management systems. The early 1990s saw the rise of relational databases such as Red Brick and Teradata, specialized or reengineered for mass updates and read-only, decision-support output. In the mid-1990s, multidimensional databases (MDDBs), most notably Essbase, appeared in response to the rigorous interactive performance requirements of OLAP applications. In the late 1990s, object database management systems (ODBMSs) made small inroads into mainstream applications.

Triggered by the growth of the World Wide Web, object DBMSs, such as Software AG's Tamino and eXcelon Corporation's Javlin, burst on the scene, optimized for storing and serving XML (eXtensible Markup Language) and for interacting with the object-oriented lingua franca of e-commerce, Java. Main-memory databases (MMDBs), such as TimesTen and Angara, entered the market as well—another solution offered for enhancing data-access performance.

The digital economy may speak new and unique languages while interacting with data stores, but most of the data it requests and receives remains quite recognizable, as that represented by the punched holes on our Hollerith's cards.

Input: From Extraction to Movement

Let's face it: Up to this point in time, data consolidation has pretty much come down to copying data from one place to another, with the goal of increasing the value, in terms of usability, of the data in the process. If the value of the data to the enterprise is not increased, the cost and effort of copying the data and storing the copy is not justified. Just the process of copying data from one place to another has a somewhat attention-grabbing (well, maybe not exactly spine-tingling) history.

Probably the most venerable example of moving data around between application systems is the GL system discussed earlier. To "feed" the GL monster, extract files are demanded from all application systems in which revenue or expense records can be found. Again under cover of darkness, the GL mon-

ster consumes these extracts, categorizing and summarizing the prior day's financial activities, then, at the end of its run, spitting out all manner of stacks of financial reports for internal management and external overseers. And the most venerated and critical of these financial management reports is, of course, the company's annual report.

A major variable in the evolution from this type of reporting system to more "authentic" decision-support applications is the breadth of the input, that is, the number of separate application sources and the diversity of the data that is consumed by the DSS. The GL systems in our prior example, while perhaps broad in the number of "feeder systems," are pretty much constrained in the types of data in which they are interested: revenues and expenses. CIFs and marketing databases (MDBs) were some of the earliest instances of extensive data consumers.

CIFs have been common in banking systems for decades. The IT landscape in banking is unique in the number of separate stovepipe applications that are typically necessary to run the business. Each banking product usually has its own specialized application. Customers typically have multiple banking products. If the supporting application systems are separate, how does a bank know that the John Doe in whose name a home equity loan is held is the same John Doe who has a checking account? Usually, the bank's CIF integrates such information.

CIFs and GL systems require periodic data feeds from multiple application systems to keep their data up to date. Early on, the computer programs required to ETL these data feeds were custom-coded by in-house IT shops. Concurrent with the growth of the data warehouse concept in the early 1990s, commercial software specialized for ETL processing began to appear. Prism Solutions was founded by Bill Inmon in 1991; Evolutionary Technologies, Inc. (ETI) was also founded in January 1991. Informatica, the current market leader, opened for business in 1993.

Data *consolidation* architectures, such as GL and CIF systems, data warehouses, and data marts, to this point had primarily followed a many-to-one model—where data from many sources is transformed and replicated into a single destination. This consolidation is typically done on a periodic, batch mode basis rather than on a synchronous real-time basis. The growth of the Web has resulted in another convergence, where transformation functionality—previously the exclusive domain of these batch-oriented ETL tools—has been combined with message brokering architectures, and from there subsumed into other families of e-commerce-focused software products: enterprise application integration (EAI) and application servers (or App Servers).

The integrated value chain (introduced in Chapter 2) required for contemporary e-commerce is likely to be supported by the same multiple, autono-

Table 1.1 Data Architecture Comparison

Property	Architecture	
	Data Warehouse Architecture	**Integrated Value Chain Data Architecture**
Synchronization frequency	Periodic (e.g., daily)	Real-time or near real-time
Source/destination model	Many-to-one	Many-to-many
Update mode	Batch	Transactional

mous transactional databases confronted by earlier data warehousing efforts. Transforming a pre-digital economy value chain into an *integrated* value chain requires data-movement, transformation, and replication technologies that provide real-time or near real-time synchronization of multiple sources with *multiple* targets—a real-time, many-to-many model. Table 1.1 compares and contrasts these two architecture types.

In response to the growing need for many-to-many, transactional, near real-time synchronization between data stores, the market for EAI software is exploding. Most EAI products bundle together several functions and technologies, including message brokering, workflow automation, and, most familiar to data warehousing technologists, data transformation and replication. EAI software products achieve data transformation and replication through message-broker middleware.

Say, for example, applications *A*, *B*, and *C* are found in various value links and channels across an enterprise. Say also that an update to the database in source application *A* must be replicated near real-time to target databases in applications *B* and *C*. EAI software enables *A* to transmit an update message containing the updated data to a central broker facility. The broker "hub" then performs any required transformations and routes update messages to applications *B* and *C*.

A class of software products labeled application servers has become highly visible since 1998. App servers, similar to EAI products, act as integration points, providing a single apparent source of data (as well as function and logic) to (usually) a Web site on the front end.

A historical overview of trends and influences in data input technologies, often referred to as *data-movement* technologies, is presented in Figure 1–3. These technologies are covered in more detail in Chapter 6.

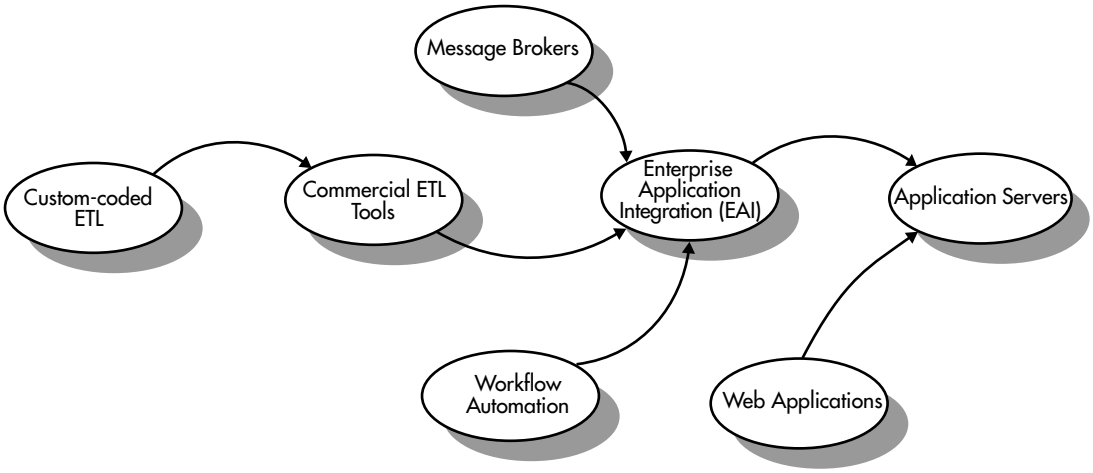

Figure 1–3
Trends and influences in data movement.

Data Warehousing and E-Commerce at Tiosa Corporation

On this, our first visit to Tiosa Corporation, we find T. Dan Roberts, Tiosa's chief information officer, up to his eyebrows in e-commerce initiatives. It seems that every business unit has already put up its own stand-alone Web site and now wants him to run it; those that haven't built their own sites are clamoring for Dan's staff to build one for them. On top of this, Collections is asking him if they should replace their electronic payment systems with XML (and what the heck is XML, anyway?), and Purchasing wants him to investigate Web-based procurement and B2B exchanges.

Dan is glad that at least the Y2K rush is over, and the various data warehouse and data mart initiatives of the past several years appear to be calming down a bit. Since he accepted the CIO position in 1995, it seems there has been one emergency after another.

Like most sizable companies, over the past three decades, Tiosa has an accumulated history of packaged and custom data warehouses, data marts, and decision-support systems. Some of the development efforts have been underwritten by technology units, some by business units; some have been successful and enduring, while others have fallen by the wayside. A proliferation of input, storage, and output technologies is the legacy of the IT decen-

tralization movement in the early 1990s. Business units for several years had wide latitude—and budgets—for investing in information technologies, and forged hearty relationships with many enthusiastic vendor representatives.

Output technologies in use within various business units include those from Brio, Cognos and Business Objects. Databases, on multiple platforms, include IMS, DB2 for OS/390, Oracle and Microsoft SQL Server. ETL tools in use in various units include those from Informatica and ETI. In the mid-1990s, a valiant data-management effort led to the implementation of the Platinum Repository MVS; the product has now fallen into disuse.

As in many companies, Tiosa's finance and marketing departments have been trailblazers in marshalling their data assets in support of improved decision making. In the finance department, a multidimensional OLAP solution based on Hyperion Essbase is used for slicing and dicing revenue and expense data that's extracted on a monthly basis from the corporate chart of accounts in PeopleSoft Financials. The risk management department is experimenting with data mining products, attempting to get a handle on large customers with whom Tiosa may have greater-than-expected exposure, and to develop profiles of customers who are likely to default on loan payments.

In Marketing, an Oracle database is used as the basis for determining the profiles of the most profitable customer segments. Custom SQL scripts are used to load the database, and a team of analysts creates and maintains reports, primarily using Microsoft Access. A PhD statistician was recently hired to investigate data mining to predict customers' likely responses to various proposed marketing strategies.

Several revenue-generating lines of business have experimented with various decision-support applications, ranging from single-user PC databases used by several remote sales offices to a product-profitability SAS application on the corporate IBM mainframe by the cash management department. Other business units, such as Mortgages and Underwriting, have yet to embark on any ambitious data-management plans outside of their core mainframe and midrange package applications.

As discussions heat up with the various departments he supports, Dan is beginning to view his challenges as twofold. First, Dan's team needs to support the development of a "coherent interaction presence" (according to the new VP of e-commerce) in the digital marketplace for the company. Second, he needs to figure out how to integrate into this digital presence the considerable internal data assets under his and others' purview. The term *architecture* continues to turn up as a recurring theme in the early discussions of this integration.

"It seems we've spent the last decade or so consolidating and redistributing data in support of all these various flavors of decision-support applications," concludes Dan in a meeting with his direct reports. "Surely we can make use

of the results of some of what we've accomplished to get data where it needs to be for these e-business projects."

Sounds like Dan may be on the right track. As we take a deeper look into the business and technical aspects of Dan's challenges, we will occasionally drop in for a status check on Tiosa's ambitious entry into the digital marketplace. Next, let's see if we can shed some light on the new VP's concept of a coherent interaction presence.