CHAPTER ONE

Defining B2B Application Integration

We are moving inexorably toward an event-driven economy, an economy where demand realized translates into demand satisfied—in a matter of nanoseconds. Demand realized is accomplished by extending the capabilities and reach of an existing IT infrastructure so that enterprise applications, both inter- and intra-enterprise, are bound by a business event–driven paradigm. In the process, expectations about business methods must be redefined. Either businesses are getting on board and automating their common business events, inter- and intra-organization, or they are preparing to exit their market. There is no in-between. This is the fundamental reality of e-Business.

This new event-driven economy has several defining characteristics.

First and foremost, it is almost instantaneous. All supplying systems are instantly aware of demand, and they are able to react to that demand instantaneously. Likewise, all demanding systems are instantaneously aware of supply capability and are able to respond accordingly.

Second, all participating systems are able to communicate in any direction, with any system, automatically, and in real time. For example, when we create an order, which in turn creates an order event, a supplier's system may respond to that event with a "delay status" because of a material shortage. The order system would learn about the delay instantaneously and just as instantly adjust the expected delivery time. As a result, customers (or more likely the customers' systems) are able to react to dynamic business realities in real time, perhaps even changing their order to work around the shortage automatically.

The benefit of this real-time, event-driven economy is obvious to anyone who has ever been frustrated when attempting to order an item from a retailer, either on the phone or online, only to discover that the retailer is incapable of determining when that item can ship. Few things are as discouraging as ordering a new car and being given a delivery date—plus or minus a month! Such a situation would be inconceivable in e-Business.

In an event-driven economy, every system affects every other system in real time. The business advantages of this are obvious: no missed delivery dates because of last-minute inventory shortages, shortages that result in valuable merchandise sitting idly on very expensive shelves. No scheduling conflicts on the plant floor because of under- or over-estimated demand. No human error or having to deal with reams and reams of paper.

Third, systems are bound at both the data and process levels. Exchanging information is not enough. Business rules, processes, and sequences need to be shared as well, ensuring that the data is processed properly and that common integrity constraints are enforced. A common business model needs to determine the path and order of each business event. For instance, the process of building, selling, and shipping a product may span as many as ten different systems housed in three different companies. A common process model needs to span the business systems, defining the properties of a business event, including the order, behavior, and characteristics of information moving from application to application, and it has to deal with concepts such as common agreements as well as private and public processes.

Finally, all relevant information existing in any participating system is accessible by any other participating system. Any information supporting any event or transaction is always available to anyone anywhere in the participating systems.

Sounds good, doesn't it? Now, all that remains is the simple question, How does your organization participate in this event-driven economy? As with most things of genuine value, the foundation must be understanding, planning, and integrating—the very things this book will explain in detail. Now is the time to begin your journey toward real-time business, something that will revolutionize your organization and every other organization that is prepared to commit to the new, challenging, exciting, and profitable economic realities that await us all.

e-Business Means Something (Different) to Everybody

The term e-Business means different things to different people and companies. To some it means Web-enabled selling. To others it means middleware. Most hold that e-Business is the next generation of Internet-enabled supply chain integration that picks up where EDI left off.

So, who's right?

No one. Everyone. The fact is, there is no clear, agreed-upon definition of e-Business. You only need to watch the IBM TV commercials to recognize this lack of consensus. "e-Business" has joined the ranks of those other popular buzzwords, such as "object-oriented," "client/server," and "Web-enablement." Everyone wants a piece of whatever it is that these terms suggest, even if few have any idea what they might actually mean. As always, there are those who crave the pure joy that comes along with buying into the current hype.

For those more determined to keep their feet firmly rooted on *terra firma,* we can say with certainty that whatever else e-Business means, it means using innovative technology to build global relationships and commerce. As such, it is the greatest opportunity and/or threat to existing business models since the industrial revolution. e-Business, that amorphous, ambiguously defined concept, is the force that is driving the fundamental reconfiguration of every existing industry.

We can also say with certainty that e-Business means more than simply doing business on the Internet. It is more than a more efficient variant of business as usual. e-Business affects nearly every aspect of business. Thanks to e-Business, customers demand more choice and convenience, brands play a bigger role, costs are dramatically lowered, and competition is bursting forth from every direction.

The finance community affords a clear example of these trends and benefits. Financial institutions are now clearing checks, trades, and loans in just minutes, not days. Manufacturing organizations are automating their supply chains in order to become more competitive and profitable.

As compelling as e-Business is as a business term, the reality is that it also refers to a cultural revolution that is taking place before our very

continued

e-Business Means Something (Different) to Everybody (*continued*)

eyes. A pleasant revolution for the most part. The explosive growth of the Internet, the proliferation of electronic commerce, sales force automation, call centers, and mobile computing, have all conspired to change the very fabric of business forever. Rather than *caveat emptor,* let the customer beware, we live at a time when the customer is becoming the single most powerful force in commerce. Relationships are changing, competition is increasing, distribution channels are exploding, and start-ups are bringing established giants to their knees.

At the risk of being redundant, we'll state it again: The value of e-Business is clear. How to make it all happen in the real world is much less clear. Bringing businesses together demands a very different technology than the one in place today. The current technology is developer-oriented. We need to advance to a business-oriented focus. We need to understand the wide range of design patterns that exist at the points of integration, and we need to understand the many scenarios for information exchange between systems, most of which may not even be under our direct control. Building e-Business systems is the biggest challenge we've faced since the first-generation mainframe systems were brought online.

Moving to e-Business

The event-driven economy is the next destination for e-Business, providing us with the ability to react instantaneously to internal or external events in order to meet consumer demand. It is also a mechanism to do business, dare we say it, "at the speed of thought."

Few examples illuminate the difference between the conventional method of doing business and e-Business more clearly than the purchase of a new car. Currently, a customer walks into an automobile dealership and orders a car. That order is then placed with the automobile manufacturer. The manufacturer in turn orders the parts and creates the car, while the suppliers order raw materials to create the parts. Paper purchase orders are sent to the suppliers, who ship the materials and send paper invoices to request payment. Only then, when all the parts are received from the suppliers, can the car be manufactured and sent to the dealer—resulting in even more paper.

This process typically takes months, not weeks. It should only take days.

Of course, most "supply chains," such as the one described in our car procurement example, are at least partially automated these days. Many owners of supply chains already use enabling technologies such as Electronic Data Interchange (EDI) to share information, such as order and payment data.

Being partially automated is not enough. We need to think more comprehensively about how we capture and react to events. We need to recognize that all components of the supply chain affect the supply chain itself. For example, when that customer walks into our car dealership and orders a car, or when that customer orders a car via the Internet, that action is a business event that is captured. Our system must react to this event by performing several tasks instantaneously: logging the event, processing the rules bound to such an event, and moving information to other interested systems or humans.

The event must be logged so that it won't be forgotten should there be a failure as it is being processed. We need to process rules bound to the event, such as price limits and credit requirements. The internal (e.g., inventory) systems and external (supplier) systems must be informed of the event. Finally, the information created by this event, in this example customer and car configuration information, must move forward to the appropriate systems. Typically, this should be a second process, or subprocess.

What is of note here is that all relevant systems are notified of the event and are supplied with all appropriate information, in real time, so that they can in turn instantly react to the event (see Figure 1.1). In our car purchase example, the sales event captured by our manufacturer's system generates an instant requirement for parts to create the car. In turn, this information triggers a cascading series of additional events within systems owned by the suppliers, events such as notifying a supplier of the raw materials required to build the parts. A single, primary event could thus trigger as many as several hundred other events, which in turn could trigger several thousand more events. It is exactly this chain reaction of events, events that serve a business need, that we are hoping to create.

Remember, this event-driven supply chain scenario is an instantaneous process. Within seconds of the initial order, the suppliers are processing requests for the raw materials, the factory floor is scheduling workers, and the logistics group is assigning resources in order to ship a car to a particular dealer. There may be hundreds of systems involved with the sale, creation, and movement of

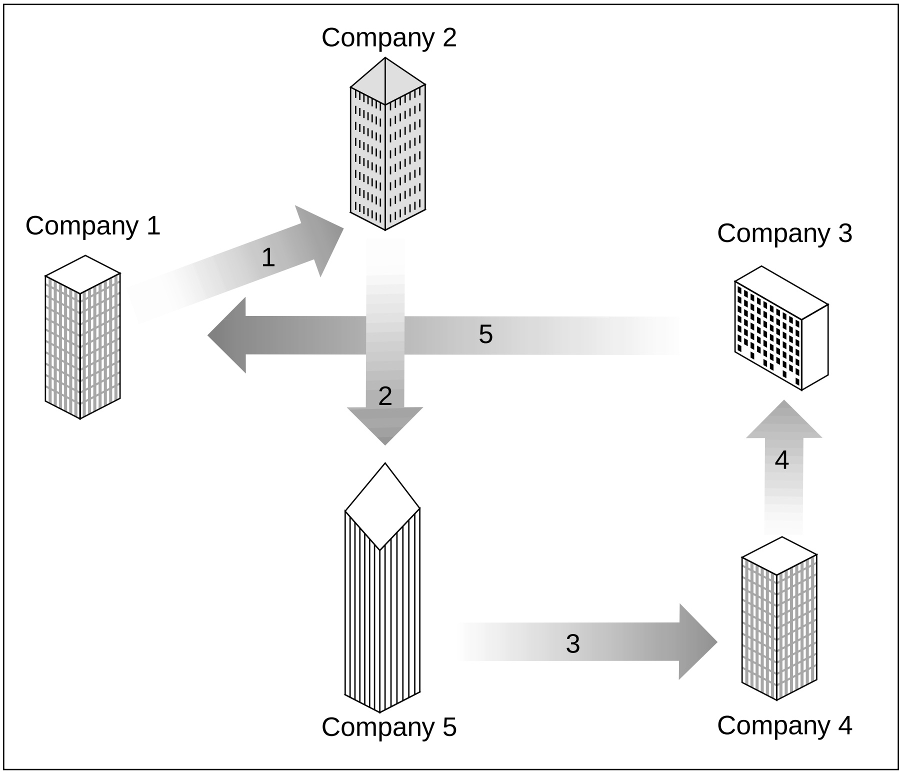

Figure 1.1 Example of an event-driven supply chain

this car, all exchanging event information simultaneously. Of equal relevance is that all systems participating in the event will be notified instantly should there be any change along the supply chain—that is, if demand changes (e.g., car sales go down) or if there is a parts shortage. Instantaneous notification is a two-way street, from orders to suppliers, from suppliers to orders.

As we learn more about integrating applications, the event-driven economy will become more of a reality. Today, systems are bound together by primitive connections, where batch and file transfers serve as the primary integration mechanisms. However, with the marriage of the Internet and more advanced middleware solutions, it won't be long before small versions of the event-driven economy are running, typically around supply chains.

This will only be the beginning. When businesses realize that this method of "speed of light" business processes means reduced costs and a huge competitive advantage, we will see the explosive growth of the event-driven economy and B2B application integration. The value of B2B application integration, or the binding of systems in support of the event-driven economy, is the topic of this book.

Using B2B Application Integration

If it is fair to assume that at some point we would like to proceed to an event-driven economy, then we need to embrace the concepts, approaches, and technology of B2B application integration. In doing so, we will once again find ourselves having to define e-Business.

As corporate dependence on technology has grown more complex and far reaching, the need for a method of integrating disparate applications between enterprises into a unified set of business processes in support of B2B e-Business has emerged as a priority. After years of creating islands of automation within each company, users and business managers are demanding that seamless bridges now be built to join these islands together, thereby allowing commerce to proceed in real time. In short, they are demanding that ways be found to bind these applications into unified cross-enterprise e-Business applications. The development of B2B application integration allows many of the enterprise applications that exist today to share both processes and data. Thus, it allows us to finally answer the demand for real-time, inter-company application integration in support of e-Business.

Interest in B2B application integration is driven by a number of factors. As the pressures of the competitive business environment move IT management to shorter application life cycles, financial prudence demands that IT managers learn to leverage existing databases and application services rather than re-create the same business processes and data repositories over and over. Ultimately, it is this financial prudence, along with the opportunity for profit, that fuels the interest in B2B application integration. The integration of applications saves precious development dollars while creating a competitive edge for those corporations that share application information either internally or with external trading partners (B2B).

Currently, the vast majority of corporations use several generations of systems that rely on a broad range of enabling technologies, technologies that have been developed over many years. Mainframes, UNIX servers, NT servers, and even proprietary platforms whose names have long been forgotten, constitute the technological base for most enterprises. These technologies, new and old, all provide some value to the enterprise, but their value diminishes if they are unable to leverage other enterprise systems that exist within other trading partners.

The case for B2B application integration is clear and easy to define. Accomplishing B2B application integration, however, is not.

What's B2B Application Integration?

So, if B2B application integration is the solution, what exactly is it? Unlike so many other buzzwords, B2B application integration is not hype dreamed up by the press and analyst community. It is, at its foundation, the mechanisms and approaches to allow partner organizations, such as suppliers and consumers, to share information in support of common business events. In short, B2B application integration is the controlled sharing of data and business processes among any connected applications and data sources, intra- or inter-company.

The challenge of B2B application integration is to be able to share data and processes without requiring sweeping changes to the applications or data structures. Unless we are able to create a method of accomplishing this integration, B2B application integration will fail to be either functional or cost-effective.

When we consider the requirements of e-Business, we must look to leverage all existing systems and bind them, within or between enterprises, to support any and all business requirements. If Company A has a sales order entry system that provides a system for recording sales, then Company B, a parts supplier, must have a parts system that is instantly integrated with Company A's sales system. Otherwise, in order to receive the parts required to produce the product sold, information about the sales must be rekeyed into the manufacturing system, which in turn produces several paper purchase orders to obtain the proper parts for the parts supplier.

Currently, Company A and Company B operate with a combination of manual and automated processes that may require days before the pieces are in place to allow the product to be manufactured. This excessive time results directly from a lack of real-time integration between all of the information systems that participate in the business event.

In order to address this problem, organizations generally move to integrate systems within their enterprise, such as the sales order entry system or the manufacturing system, as in our earlier example. However, such a "solution" only addresses a piece of the puzzle. Unless the parts-supplier systems (Company B) are tightly integrated with the sales order entry systems (Company A), the time between the order and the delivery of the product will continue to be excessive.

A business transaction takes at least as long as the slowest system in the loop. It is easy to imagine a well-integrated set of systems within an organization, systems that are able to share information in real time. However, unless that organization's trading partners also participate in the real-time exchange of information (B2B integration), the trading partner's latency will limit the efficiency of the organization. As a result, competitiveness and profitability will suffer.

B2B application integration extends EAI to include access to perfect information on demand to outside trading partners, enabling them to react instantly to a business event. To play off our previous scenario, integrating all information services means that an entry in the sales order entry system would automatically and instantaneously trigger events in the manufacturing systems, events that would in turn trigger events at all relevant trading partners, including the parts supplier (see Figure 1.2).

Leveraging Your Assets

Fundamentally, B2B application integration is about leveraging existing systems and databases (yours or your trading partners') by allowing them to communicate seamlessly in support of a business purpose. We want to integrate your current systems and avoid having to create new applications. As we will see, however, integrating systems that were typically built without integration in mind can be something of challenge.

Even as the technology supporting your applications has aged, the value of the applications to your enterprise likely remains fresh. Indeed, your "ancient" technology has probably remained critical to the workings of your enterprise. Therefore, access to those systems remains valuable to you and to your trading partners as well.

Unfortunately, many of these business-critical systems are nearly impossible to adapt so that they are able to communicate and share information with other systems. There is, of course, the option of simply replacing these older systems. However, the cost of doing so is generally prohibitive. In any case, the decision to do so may very well be beyond your direct control.

Packaged applications such as SAP, Oracle Financials, and PeopleSoft—which are natural stovepipes themselves—have only compounded the problem. Sharing information among these systems is particularly difficult because many of them were designed not to access anything outside their proprietary technology.

Applying Technology

If B2B application integration lays bare the problem, then traditional middleware seeks to reveal the solution—sort of. Unfortunately, traditional middleware only

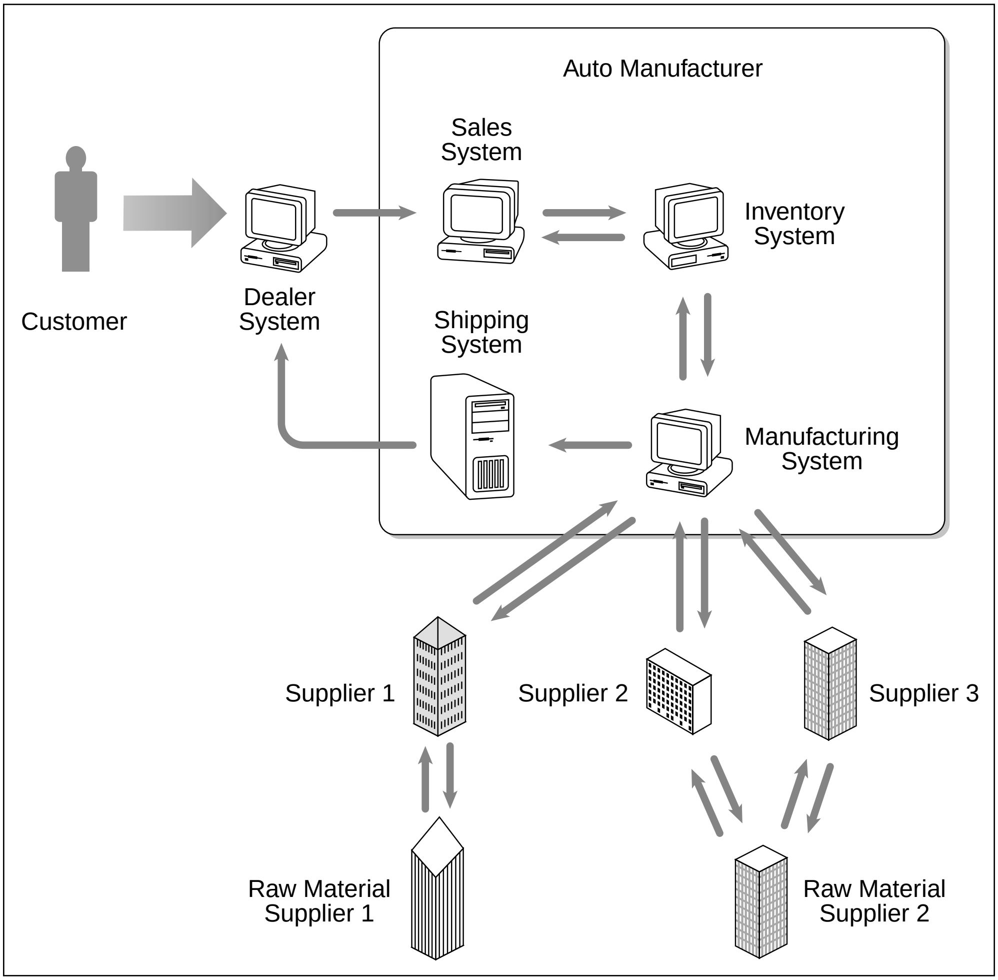

Figure 1.2 Using B2B application integration to leverage the value of all systems in a trading community

addresses the B2B application integration problem in a limited manner, primarily because traditional middleware was built to integrate applications within an enterprise and does not take into account the special needs of B2B integration.

The primary limitation is that the middleware that uses message queuing or remote procedure calls (RPCs) only provides point-to-point solutions—that is, a link between system A and system B. Unfortunately, any attempt to link additional systems rapidly deteriorates into a complex tangle of middleware links, both intra- and inter-company (see Figure 1.3). Even more troubling is that traditional middleware requires significant alterations to both the source and target systems, embedding the middleware layer into the application or data store. Changing the source and target systems to support e-Business may be out of the question, because an organization rarely controls the systems of its trading partners.

For example, in attempting to integrate a custom accounting system running on Windows 2000 with a custom inventory control system running on a mainframe within another company, you may select a message-queuing middleware product. Such a middleware product will allow both systems to share information—over the Internet, for example. However, because a point-to-point middleware layer only provides a program interface, you generally have to alter the source system to make it understandable to the target system. This is costly and sometimes risky. Unfortunately, it is also sometimes impossible.

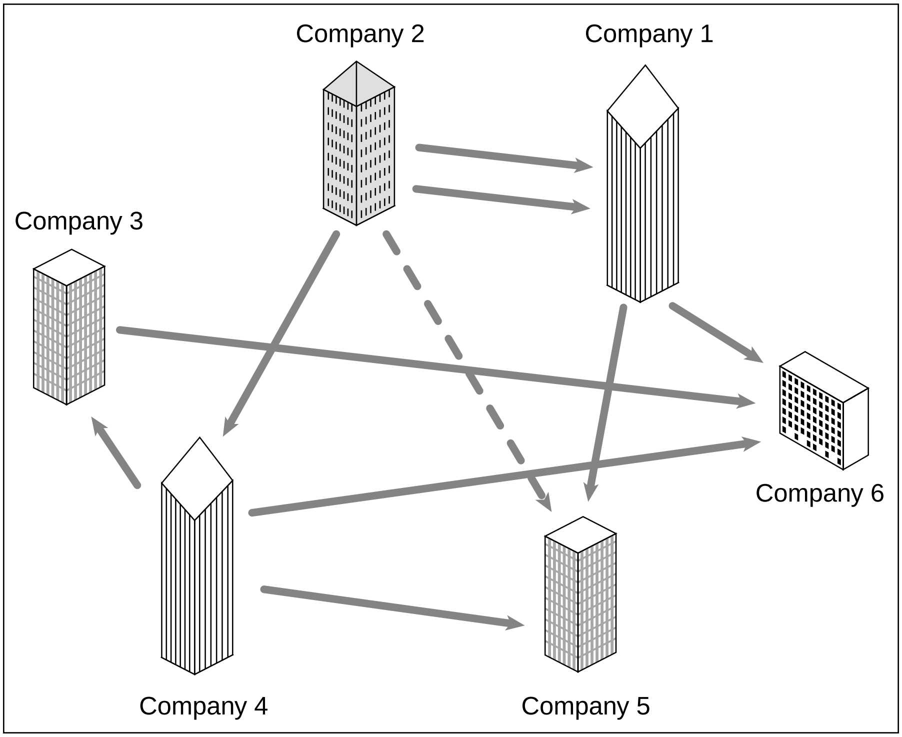

Figure 1.3 Point-to-point integration solutions are common but dysfunctional for B2B application integration.

As we use the same, or similar, technology to integrate other applications inside or between companies, the number of point-to-point solutions must grow to accommodate the increased information flow between various systems. The end result is a complex and confusing maze of software pipes running in and out of existing enterprise systems without any central control or management. As a consequence, this scenario results in a limited ability to react to change and a strategic value that is minimal at best.

This scenario is complicated by the demand on IT managers to perform integration projects inside fluid environments, using rapidly advancing technology.

As if these structural and strategic limitations weren't enough, the economics of traditional middleware have placed B2B application integration beyond the reach of most organizations. According to the Aberdeen Group, even a simple, dual-application linking is financially daunting, running as high as $10 million. Given these significant limitations, it follows that B2B application integration demands a very different method of application integration from one that relies on traditional middleware solutions.

The requirement of B2B application integration creates its own set of integration-level application semantics. Stated another way, B2B application integration creates a common way for both business processes and data to speak to one another across applications and between linked organizations. Although the problem of inter-process and inter-data communication is an old one, we approach it with a new set of technologies, technologies designed specifically for B2B application integration.

Keeping all this in mind, we can focus on the following differences between traditional approaches and the vision of B2B application integration:

- B2B application integration focuses on the integration of both business-level processes and data between organizations. The traditional middleware approach was data-oriented and was typically a simple, intra-company application-to-application solution.
- B2B application integration includes the notion of reuse in addition to distribution of business processes and data between several linked enterprises.
- B2B application integration is an application-to-application concept, always functioning in near real time and typically at the back-end with limited end-user influence.
- B2B application integration allows users who understand very little about the details of the applications to integrate the applications.

- B2B application integration works on the notion of common agreements between trading organizations and supports those agreements as information is exchanged between them.
- B2B application integration assumes that most source and target systems cannot be altered to support B2B application integration, thus the points of integration must be nonintrusive.
- B2B application integration takes into account the differences between integrating applications within and between enterprises, and supports a single process model that spans both.
- B2B application integration must take advantage of advanced security standards to ensure that information moving between companies is not visible to others on the public networks.

Making the Business Case for B2B Application Integration

We have already noted that the business environment no longer supports using technology for technology's sake alone. To justify its expense, a technology must demonstrate its usefulness. The technology supporting B2B application integration is no exception. Although the case for B2B application integration is clear to most people familiar with the technical aspects of this discussion, it might not be as clear to others who also need to understand its value. For example, will implementing B2B application integration between several enterprises provide a return worthy of the investment? If it will, how long will it be before the return is realized? Is B2B application integration a short-term or long-term proposition? Perhaps most important, what are the methods that best measure success?

In establishing an argument for B2B application integration, we should understand a number of things. First, implementing B2B application integration requires that at least one person in the organization thoroughly understand the business processes in the enterprise and in the enterprises of partner organizations. This knowledge determines the degree of integration necessary to optimize those business processes. Although methodologies and procedures exist that can be applied to the task, most competent managers understand the degree of value when applying B2B application integration without over-analyzing this information.

Not all organizations are equal. Some organizations will benefit more than others from B2B application integration. Some organizations clearly require a B2B application integration initiative, while others might find little value in implementing B2B application integration within their enterprises. Although it may be possible to develop a common-sense method to evaluate the potential success of B2B application integration, the reality is that “success” must be defined on a case-by-case basis to account for the many factors that exist in any given e-Business problem domain. Unfortunately, there is no broad-based definition of B2B application integration success. Success must be measured “one enterprise at a time.”

To evaluate the potential value of B2B application integration to your enterprise, you must establish a set of measures that define success for your organization. You can accomplish this by examining and measuring the current state of the enterprise. With this baseline, consider your goals and the effort that will be required for you to realize those goals. For example, if increasing sales is one of your goals, sharing inventory information with the sales order-processing system and the parts-supplier systems of your trading partners may help you realize your goal. Even so, this integration effort will have minimal value without

EAI Meets e-Business

At this point, you may be confused as you try to understand the differences between EAI and B2B application integration. At first glance, these differences *are* confusing. In a real sense, they share many of the same approaches and technologies. This book bears that out.

EAI typically deals with the integration of applications and data sources within an enterprise to solve a local problem. In contrast, B2B application integration is the integration of systems between organizations to support any business requirement, such as sharing information with trading partners (see Figure 1.4). Although EAI and e-Business exist in different problem domains, the technology and approaches applied to both EAI and e-Business solutions are similar. For example, both may employ middleware solutions, such as message brokers, to exchange information between various systems. Additionally, approaches to integrating systems can be very much the same. Within most e-Business problem domains, EAI solutions should come before B2B application integration. Logic suggests that you must be able to integrate your internal information systems before they can be externalized to foreign systems residing within your trading partners. EAI and e-Business are clearly integrated concepts that leverage much for each other.

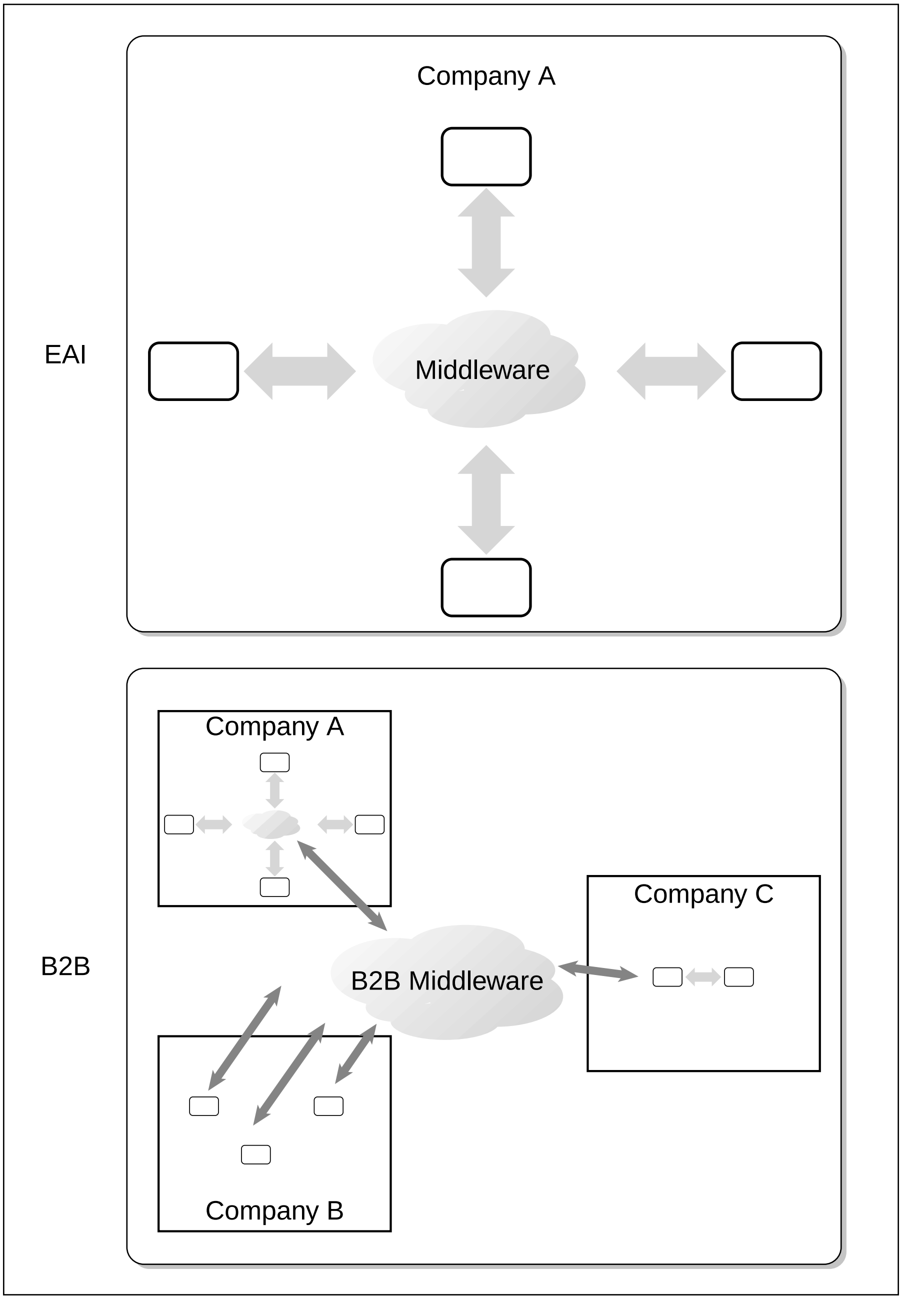

Figure 1.4 **EAI and B2B application integration differ in both problem domains and approaches.**

a concurrent and significant gain in user productivity or a reduction in error rate. In order to accurately assess the potential benefits of B2B application integration to your organization, you must assess both user productivity and error reduction. Only then can you determine the potential impact of B2B application integration on your enterprise's bottom line.

Middleware and B2B Application Integration

We will devote a great deal of our attention to middleware technology. In the context of B2B application integration, middleware is used as a simple mechanism to move information and shared business logic between applications. In short, it is the underlying technology of B2B application integration.

As we have previously noted, application integration, using a broad range of connection technology, has existed for years. In the past, application integration was low-level play, with developers working at the network protocol layer or just above, before advancing to true middleware solutions, such as RPCs, MOM, and transactional middleware (see Figure 1.5). Now, the next generation of middleware has arrived, with new categories such as message brokers, B2Bi servers, application servers, distributed objects, and intelligent agents. It is reasonable to expect more middleware and middleware categories to emerge as interest in B2B application integration grows.

Middleware is essential in that it hides the complexities of the underlying operating system and network in order to facilitate the integration of various systems in the enterprise. In most cases, developers deal with an API on each system. The middleware is responsible for passing information through the different systems on behalf of the application. These APIs are general-purpose, data-movement or process-invocation mechanisms. In most cases, they do not know the applications and databases they are tying together. Traditionally, developers have had to create the links for the middleware. However, we are moving rapidly to a "plug-and-play" solution set in which the middleware will integrate applications intelligently, with little or no programming.

Middleware provides developers with an accessible way to integrate external resources using a common set of application services. These external resources may include a database server, a queue, a 3270 terminal, Enterprise Resource Planning (ERP) applications, a custom API, or access to real-time information. In the world of distributed computing, middleware provides a means to connect clients to servers, clients to clients, and servers to servers without the need to navigate through many operating systems, networks, or resource server layers.

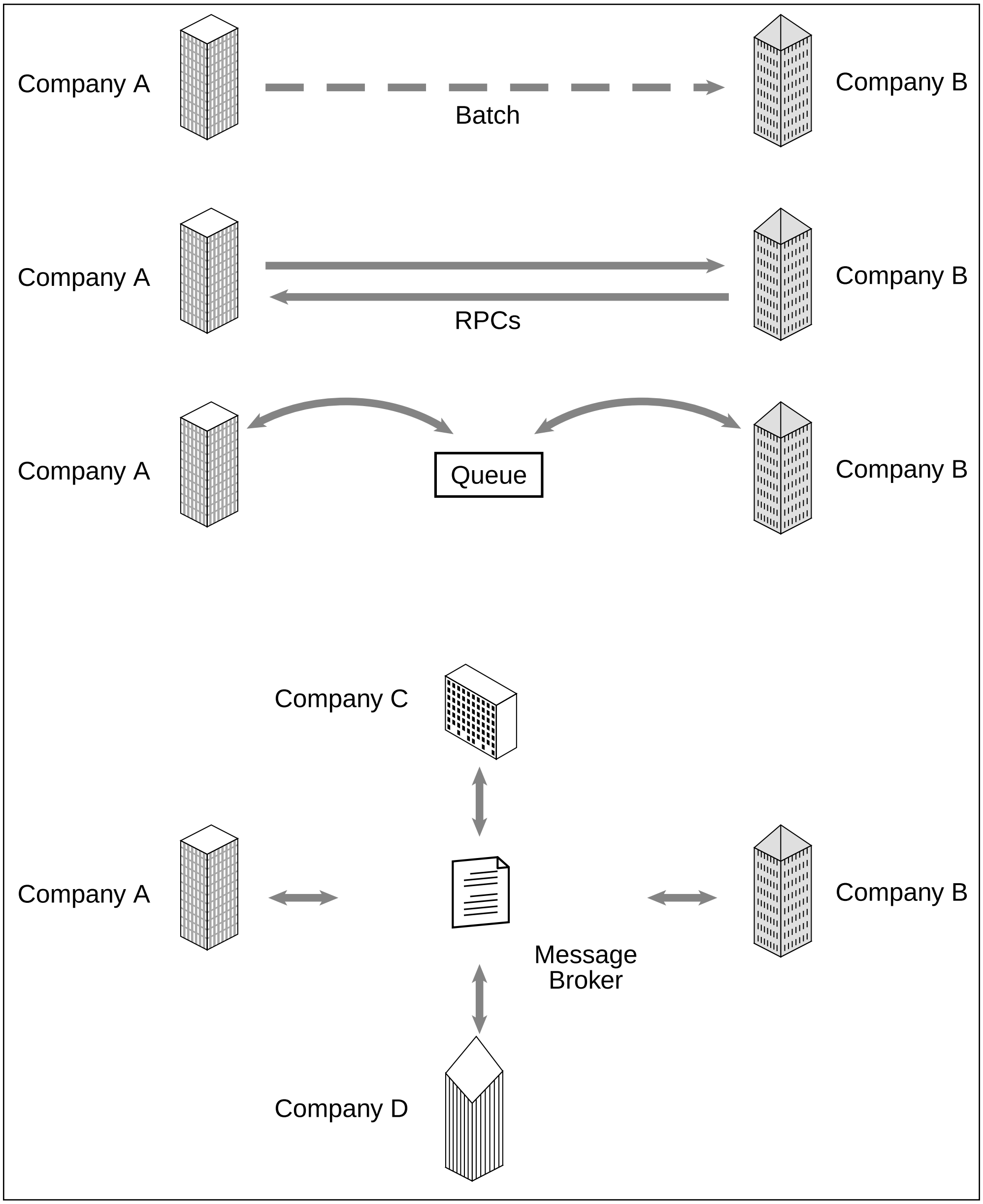

Figure 1.5 **Over time, middleware has evolved to support more sophisticated solutions.**

Middleware Makes B2B Application Integration

Middleware, once only for use intra-enterprise, has now become an enabling method for B2B application integration. Most major middleware vendors, including BEA, SAGA Software, IBM, NEON, Tibco, and even Sun, are reworking their products to address B2B application integration. It seems as if every vendor that can connect application A to application B is positioning itself to be able to claim dominance in the emerging e-Business marketplace.

Middleware is moving in the direction of e-Business. In many cases, this amounts to little more than "teaching an old dog a new trick." We are learning that the same technology that binds applications within enterprises can bind applications between enterprises as well. However, we are also learning that simply redirecting traditional middleware toward the new B2B problem domain is not going to be sufficient by itself to do the trick. Indeed, middleware is retooling for B2B, adding new features such as:

- Support for inter- and intra-process integration
- Support for B2B standards, including RosettaNet, ebXML, and EDI
- Support for Internet-enabled information exchange
- Support for advanced security models

Approaching e-Business

There is a certain bemusement that accompanies watching traditional and new middleware vendors approach e-Business. Each clearly believes that its technology represents the best approach to providing customers with the "most" of something—whether that something is flexibility, scalability, reliability, usability, and so on. Although "the best approach" is not always easy to assess, it is clear that each vendor attacks the e-Business problem in its own special way.

Although it's difficult to pigeonhole the e-Business approaches that middleware vendors are taking, we can create some general categories. These categories include business rules integration, information integration, process integration, and collaboration (see Figure 1.6).

Business rules integration refers to the binding of application logic between two or more e-Business partners. This means that the composite applications that exist are accessible to all interested parties. As a result, the exchange of both information and business rules is fully automated.

Although all vendors seem to claim that they provide the infrastructure for sharing rules and data, the application server and distributed object vendors

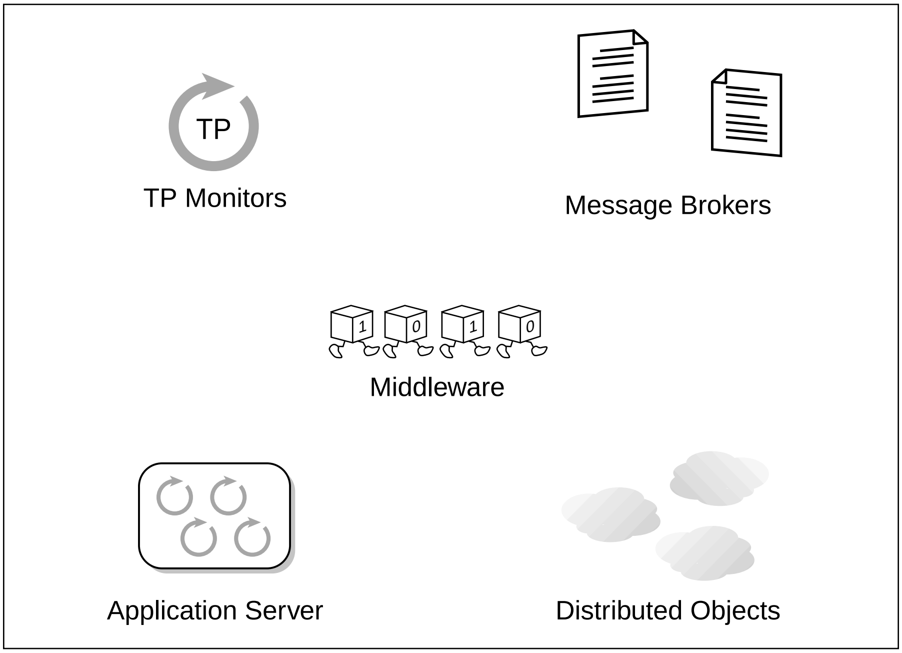

Figure 1.6 Categories of middleware

actually provide clear business rules integration approaches. They allow developers to rehost, or expose, existing rules or methods to other applications that may need them to support a virtual e-Business system. For example, many banks have established the Common Object Request Broker Architecture (CORBA) as a standard mechanism to share application services both within and between banks.

The **information integration** approach to e-Business enablement operates at a slightly lower level than business rules integration. It provides a platform for exchanging relevant business data in order to support e-Business initiatives—for example, the exchange of order and invoice data with a customer and vendor (see Figure 1.7).

Information integration is relatively inexpensive and generally does not require many changes to the participating systems. As such, it is often the first step with most e-Business projects. Message brokers, data replication engines, and data migration engines all take the information integration approach. The Extensible Markup Language (XML), which provides a common information exchange format for many incompatible applications and data sources, has been providing the most value in this context.

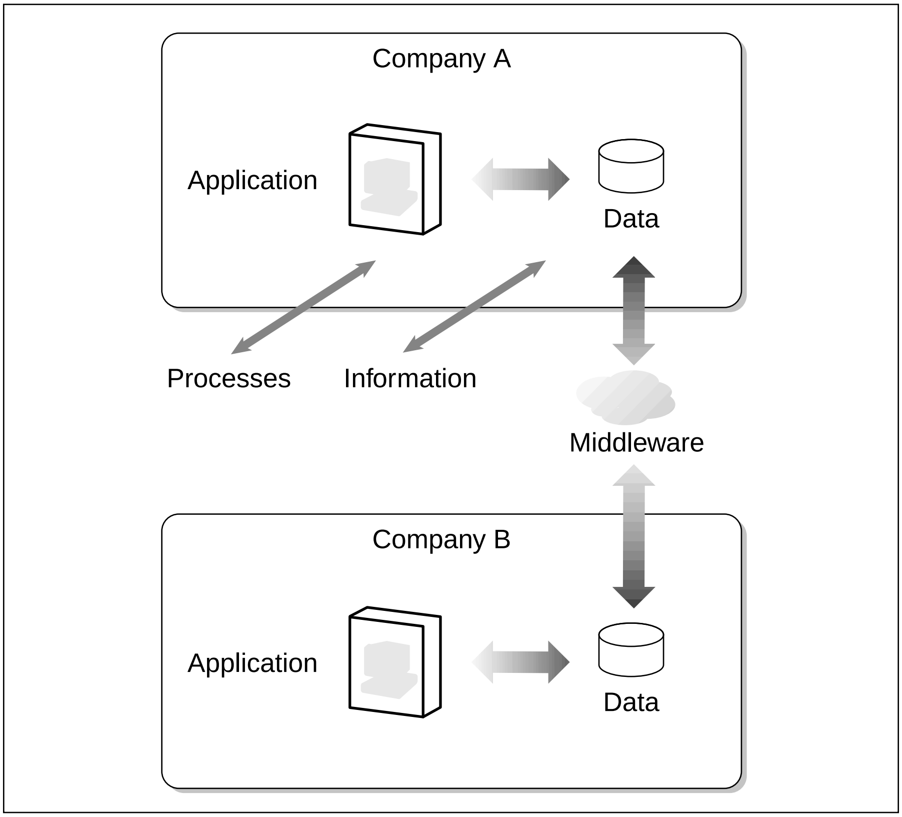

Figure 1.7 Information integration approach to B2B application integration

Process integration to e-Business enablement provides a set of processes that function above both business rules and information integration. Process integration is unlike traditional middleware. It is in actuality a process model that resides on top of middleware and provides both logical and physical information flows over existing business systems (see Figure 1.8).

Process integration is important to e-Business because it provides an abstract business layer that exists over the physical plumbing, a layer that allows process integration tool users to map out the logical flow of information between systems within the same enterprise.

Collaboration middleware provides a “soft touch” approach to information movement. Collaboration typically means providing a geographically dispersed

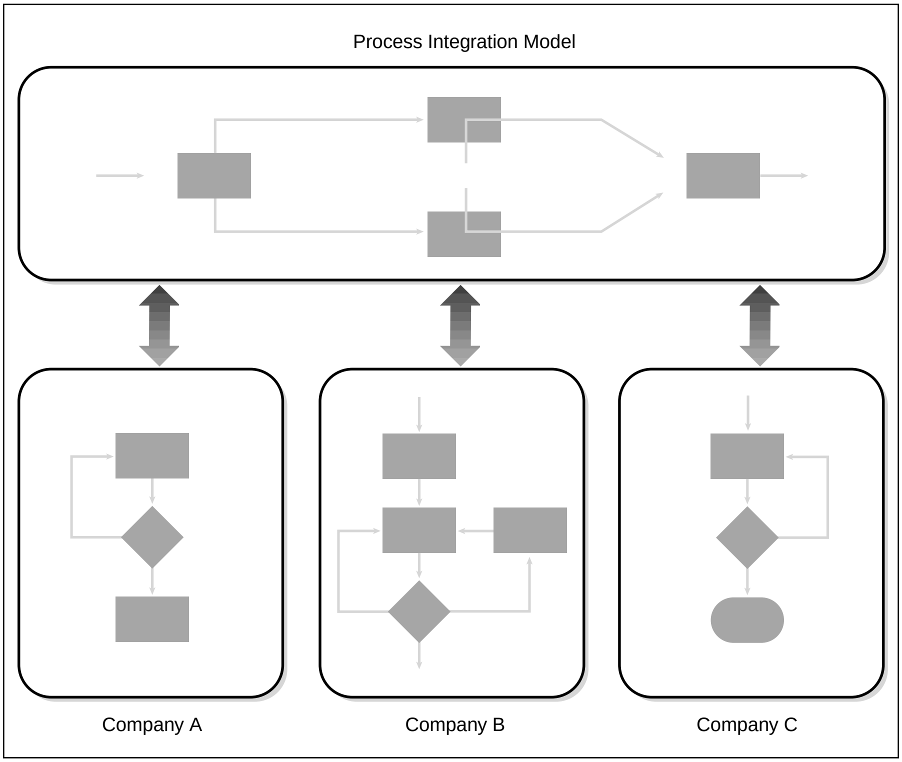

Figure 1.8 Process integration approach to B2B application integration

workgroup with the opportunity to share messages and other information in real time to support a business need (see Figure 1.9). Applications include customer relationship management, online customer service, and virtual product development.

Collaboration's great strength is its ability to support virtual communities of participating humans and computers. As a result, the middleware must be "human-aware" and capable of providing an interface to other humans and systems. It must also be data- and information-aware and capable of providing information to anyone from anywhere. Collaboration uses a centralized set of middleware to manage the movement of information. Collaboration and process integration approaches share many of the same design patterns. Throughout this book, we'll typically bind them into one concept—collaboration. (See the section Types of B2B Application Integration later in this chapter.)

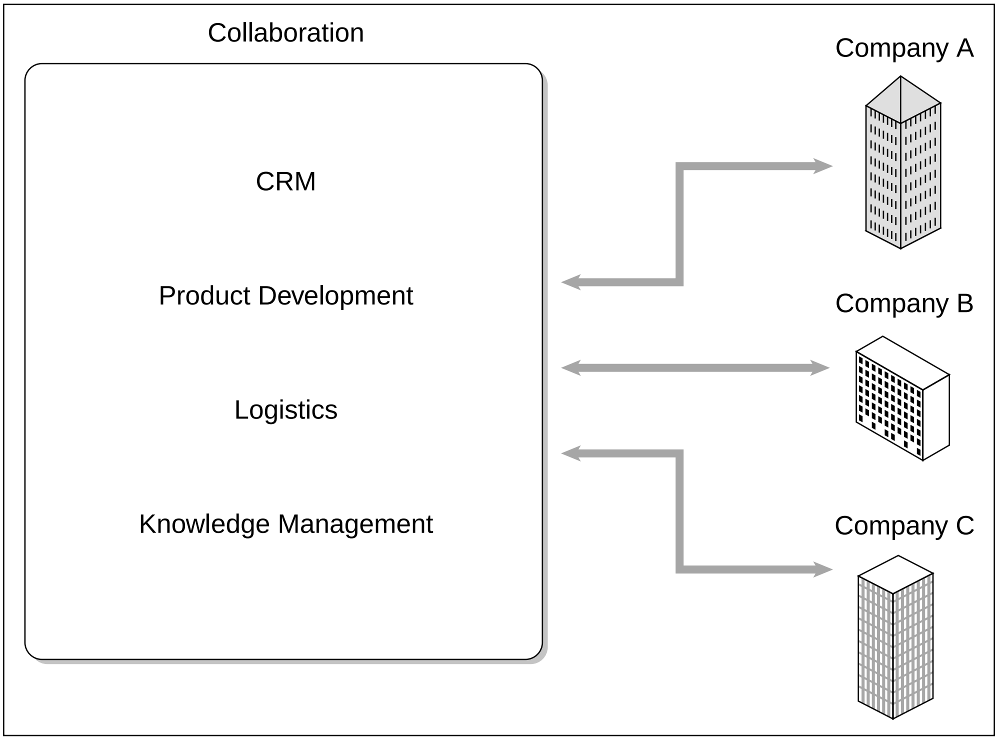

Figure 1.9 Using collaboration to approach B2B application integration

New e-Tricks for Old Dogs?

Is it reasonable to suggest that traditional middleware will provide the proper services for the new e-Business problem domains? Perhaps . . . in some cases. Traditional middleware still needs to add many more features if it is going to learn to do everything it needs to do.

In order to move forward, existing middleware vendors need to understand the various design patterns of the machine and the human entities they are interacting with. For the most part, traditional information movement applications utilized an API and employed synchronous and asynchronous mechanisms to communicate with systems. Unfortunately, APIs may have little value in the new world of e-Business. In most cases, changing the systems that exist within other enterprises in order to solve an e-Business problem is not an option. Instead, e-Business systems must interface with source and target systems through innocuous points of integration. This might require extracting data from a database or dynamically reading Web pages. More to the point, e-Business–enabled middleware must understand how to use the Internet as its primary mechanism of communication. It must also learn to deal with such pesky realities as firewalls and bursty performance.

Additionally, middleware vendors must consider the type of interface to be employed. Traditional synchronous request-reply interfaces are much too primitive for a typical e-Business problem domain. B2B-enabled middleware must take a more event-driven approach. B2B interactions are bound to challenge the capabilities of today's static interfaces. New, more dynamic and intelligent mechanisms need to be devised to deal with systems that may not be under the control of the e-Business system owner.

We may have to interact with systems whose only interface exposes data through a set of Web pages. If the pages change, which they always do, the interface must be able to react to the change without requiring redevelopment. When XML becomes more of the standard, these types of interfaces will present less of a challenge. However, XML itself will inevitably bring its own set of complexities and level of sophistication.

Finally, information movement mechanisms, which include brokering and transaction processing, have been the traditional choice to provide middle-tier processing capabilities. Within the e-Business problem domain, we may have to mix and match the two, providing brokering capabilities (e.g., translation, routing, and rules) along with application and user interface processing. Therefore, transactional middleware needs to incorporate the basic features of a message broker, and message brokers must incorporate the basic features of transactional middleware, in order to provide the maximum amount of value to e-Business systems (see Figure 1.10).

To illustrate this point, let's consider that an application server might do a good job of externalizing information from many different systems through a Web interface. However, the application server's strict use of transactional semantics limits its ability to operate independently of the remote resources (e.g., queue, database, packaged application) it is interacting with. Message brokers are very good at moving information from place to place and do so independently of the source and target systems, but they are not particularly good at externalizing back office information through a user interface.

e-Business development demands the best of both worlds.

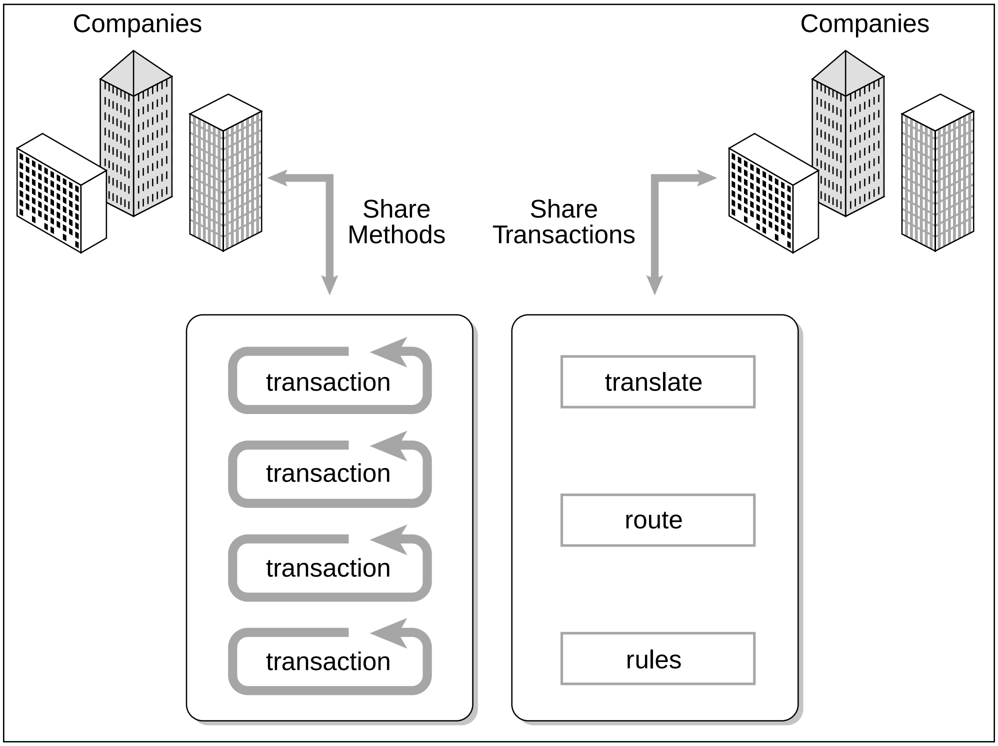

Figure 1.10 **In the world of B2B application integration, message brokers and transactional middleware must learn to work together.**

Types of B2B Application Integration

When contemplating B2B application integration for your organization, you must first understand the sum and content of the business processes and data in your organization. IT also needs to understand how these business processes are automated (or not automated, as the case may be) and the importance of all business processes. Depending on your enterprise, this effort may demand a significant amount of time and energy. Many organizations seek new methodologies to assist them in this process and use the opportunity to examine the best practices.

In brief, organizations must understand both business processes and data. They must then use this understanding to determine which processes and data elements require integration. This process can take on several dimensions (see Figure 1.11), including:

- Data-oriented
- Application interface–oriented
- Method-oriented
- Portal-oriented
- Process integration–oriented

Although this book includes chapters devoted to each type of B2B application integration, a brief overview of the various types is provided here.

Data-Oriented

Data-oriented B2B application integration is the process—and the techniques and technology—of extracting information from one database, perhaps processing that information as needed, and updating it in another database within another organization. Although this process may sound simple and straightforward, in a typical B2B application integration–enabled enterprise it might mean drawing from as many as 100 databases and several thousands of tables. It may also include the transformation and application of business logic to the data that is being extracted and loaded.

Cost is the primary advantage of data-oriented B2B application integration. Because we are mostly leaving the application alone and not changing code, we don't need to incur the expense of changing, testing, and deploying the application.

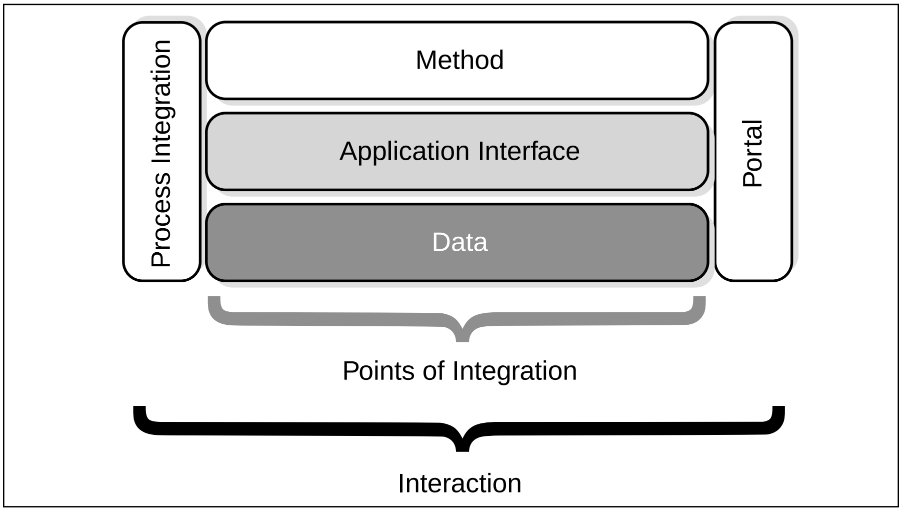

Figure 1.11 Dimensions of B2B application integration

What's more, the technology that provides mechanisms to move data between databases and reformats that information is relatively inexpensive when compared to other B2B application integration levels and their applicable enabling technology.

Application Interface–Oriented

Application interface–oriented B2B application integration refers to the leveraging of interfaces exposed by custom or packaged applications. Developers leverage these interfaces to access both business processes and simple information. Using these interfaces, developers are able to bundle any number of applications, allowing them to share business logic and information. The only limitations to this strategy rest with the specific features and functions of the application interfaces.

Application interface–oriented B2B application integration is most applicable to packaged applications, such as SAP, PeopleSoft, and Baan. These applications expose interfaces into their processes and data but do so in very different ways. In order to integrate those systems with others in the enterprise, we must use these interfaces to access both processes and data, extract the information, place it in a format understandable by the target application, and transmit the information. Although a number of different technologies can do this, message brokers seem to be the preferred solution.

Method-Oriented

Method-oriented B2B application integration is the sharing of the business logic that exists within the enterprise. For example, the method for updating a customer record may be accessed from any number of applications, within or between organizations. These applications may access each other's methods without having to rewrite each method within the respective application.

There are numerous mechanisms for sharing methods among applications. These include distributed objects, application servers, Transaction Processing (TP) monitors, frameworks, and even creating a new application that combines two or more applications.

There are two basic approaches to combining applications.

1. Create a shared set of application servers that exist on a shared physical server, such as an application server.
2. Share methods already existing inside of applications using distributed method-sharing technology, such as distributed objects.

Method-oriented B2B application integration is something we've been practicing for years as we sought to reuse application development efforts within the enterprises. We've not been largely successful because of both human and technological issues. Perhaps with B2B application integration, we may get it right.

Portal-Oriented

Portal-oriented B2B application integration is very popular today thanks to the mushrooming use of the Internet. Using this approach, application architects can integrate applications by presenting information from several local or partner applications within the same user interface.

Those of us who use Web portals such as www.excite.com, www.yahoo.com, or www.snap.com daily are already familiar with this concept. Information from many places, such as other sites or applications, is presented within the same user interface, typically a Web browser. Enterprises are avoiding the complexity and expense of traditional back-end integration by leveraging this integration approach as a means of integrating enterprise systems (such as inventory, SAP, and sales automation systems from the earlier example) at the user interfaces.

Process Integration–Oriented

Process integration–oriented B2B application integration, at its core, is a sophisticated management system that places an abstract business-oriented layer on top of more traditional B2B information movement mechanisms. Process integration–oriented e-Business provides those who are supporting B2B application integration with a business-oriented and process automation–like view of how business information flows between trading partners. Collaboration-level B2B application integration does not typically deal with physical integration flows and physical systems but with abstract and shared processes such as people, invoices, orders, companies, and merchandise. An example of process integration–oriented B2B application integration is integration that provides a common abstract process between trading partners to support the development, construction, and delivery of durable goods, such as an automobile.

Process integration–oriented B2B application integration is at the top of the food chain in the world of B2B application integration, leveraging other types of B2B application integration, including data-oriented, method-oriented, and application interface–oriented. This is an emerging concept but one that will ultimately drive most B2B application integration activity going forward, once the physical integration problems are solved.

In Chapters 2 through 6, we'll take a closer look at these types of B2B application integration as well as the enabling technology that supports these concepts.