

# 1

---

## Event Processing Overview

This chapter introduces you to events, event-processing systems, and event-driven architecture. You'll learn about the relevance of event processing to contemporary management strategies and begin to see how event processing can improve a company's timeliness, agility, and information availability.

### Introduction to Events and Event Processing

An *event* is anything that happens. A *business event* is an event that is meaningful for conducting commercial, industrial, governmental, or trade activities. Examples include receiving a customer order, making a bank payment, experiencing a power outage, changing a customer address, suffering a network security breach, detecting signs of attempted fraud, hiring an employee, and spotting a change in a competitor's price. Events small and large take place all day, every day in every corner of a company and its environment.

A fundamental characteristic of events is that they cannot be entirely foreseen—a company doesn't know in advance when an event will occur or the details of the event's nature. Customers place orders, competitors change prices, and dishonest people attempt fraud according to their own wants and needs, not on a schedule or plan known to the company or its employees.

A company, person, or animal is said to be *event-driven* when it acts in direct response to an event. The event acts as a stimulus, triggering some reaction. Some events represent threats that must be addressed. For example, a zebra that encounters a lion on the savannah will be event-driven to run away as fast as possible. Or a bank that picks up signs of credit card fraud will be event-driven to stop accepting charges on that card immediately.

Other events are positive opportunities that can be exploited. For example, a sales manager who receives a report that snow blower sales in the New York region are exceeding expectations will be event-driven to order an extra shipment of snow blowers to take advantage of the market conditions. A trader who detects a price differential for a certain commodity in two different markets will be event-driven to buy in one market and immediately sell in the other to profit from arbitrage.

Every company or person exhibits event-driven behavior some of the time, but many activities are time-driven or request-driven. A typical corporate business process has event-driven, time-driven, and request-driven aspects. Event-driven behavior is the best approach when dealing with unpredictable factors and situations. The per-

son or system who responds to an event may be guided by a general policy or standard operating procedure but might not know when to implement the policy or procedure or exactly how to apply it to a particular situation until the actual event is detected. Time-driven behavior occurs when the nature and timing of an activity can be planned in advance. Request-driven behavior is appropriate when the nature of the activity is understood and jointly agreed on by multiple parties but the timing is not predictable.

Because the world is full of events, airplanes need pilots, not just flight plans. A flight plan can describe a trip in advance, taking into account landmarks, distances, weight, historical data on fuel consumption, weather forecasts, and reported wind velocity. But a pilot is required to carry out the plan and make ongoing adjustments, as the plane will inevitably encounter other planes and changes in the weather, and perhaps have mechanical problems or experience other unforeseeable conditions while en route to its destination. Most of a pilot's job is event-driven. Drones and automatic pilots have become practical in recent years precisely because of the emergence of the technology described in this book. Our primary focus is on commercial business systems rather than on smart devices but the principles are the same.

All companies and people pay attention to events because the real world is complex and dynamic, and no one has complete foreknowledge of what will happen. Activities that are strongly affected by external factors—things that are outside the control of the company—are particularly suited to event-driven behavior. For example, field sales and service operations, customer contact centers, web-based sales and marketing, other customer-facing activities, supply chain management, transportation operations, and anything to do with competitive markets are fertile ground for event-driven behavior and event-processing systems. Internal processes that are more under the control of the company have more time-driven and request-driven aspects, but even they are sometimes event-driven because unexpected circumstances arise everywhere.

Being event-driven in a real-world sense is clearly nothing new. If your company is advised to become an event-driven enterprise, that doesn't mean that your company should start handling events—your company already does that. What it means is that your company should improve *how* its business processes and IT systems handle events. Too often, companies respond to events by using outdated processes that were designed for a slower-paced, less-automated world. They don't take full advantage of the wealth of event data that is available in enterprise application systems, on the Web, or from other potential sources of event data. People making minute-by-minute, operational decisions often don't get information on current events soon enough. In other cases, people are overwhelmed by the volume of event data that is sent their way so they miss the truly important observations. Most companies could streamline, simplify, accelerate, and improve operations in hundreds of ways by applying the discipline of event processing.

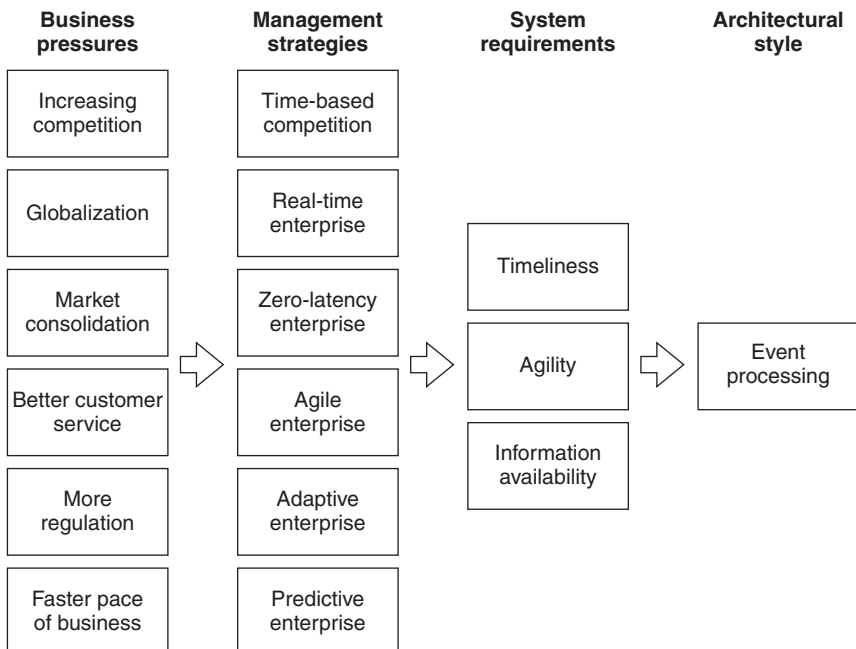
## **Business Context**

Companies use the discipline of event processing to improve business performance, not to make their IT departments run better (although event processing can some-

times do that, too). Today's companies face escalating demands, as shown in the Business Pressures column in Figure 1-1.

Competition is growing as new companies emerge. Globalization is also increasing the number of competitors that any business is likely to encounter in a traditional geographic market. Many markets are consolidating as weaker players lose ground to stronger companies. Companies strive to outdo each other by providing faster and better customer service, new products, and, in many cases, lower prices. Customer expectations keep rising, putting pressure on every company to improve its operations. When someone applies for a loan or an insurance policy, she expects a quicker decision and more visibility into the process than was available in earlier days. When a customer submits an insurance claim, requests a repair service, or ships a package or letter, he wants quick action and may want to know the status of his work item at all times. When he places an order online, he expects that the goods will arrive sooner than in the past and wants a tracking number so he can anticipate the delivery time.

New requirements also arise from external regulators and internal decision-makers. Governments and other official bodies impose an ever-growing amount of regulation that requires more reporting and better transparency of operations. Executives, managers, and operational decision-makers on every level want fresher and better information from dashboard displays, e-mail, and other forms of alerts so they can have a clearer picture of what is happening. Operational control activities, such as



**Figure 1-1:** Business pressures affect application architecture decisions.

supply chain management, are far more effective today because of the increasing availability of current information about events. Technology makes it possible for companies to respond to all of these requirements, but it also encourages these requirements to grow even more over time.

These business pressures have inspired the development of numerous modern management strategies (see Figure 1-1, Management Strategies column). Time-based competition, real-time enterprise, and zero-latency enterprise strategies highlight the benefits of timeliness. Agile enterprise and adaptive enterprise strategies focus on the importance of flexibility and change. Predictive enterprise strategies emphasize a combination of information availability and timeliness. Each strategy has its own set of truths and benefits, but they all ultimately depend on improving a company's timeliness, agility, and information availability (see Figure 1-1, System Requirements column) to varying degrees. For example, fast action isn't much good if you are doing the wrong things quickly, so you need good information as well as timeliness. Doing the right thing quickly is good, but you also need agility so you can adjust your behavior when the business requirements change, because the "right thing" will change over time.

This is not a book on business architecture, so we won't dwell on business pressures and management strategies. We mention them here to explain why people pursue event processing. For the purpose of this discussion, we assume that you and others in your company already know which of these business requirements and management strategies apply to your situation, or you can discover them by other means. We also take for granted that you appreciate the value of timeliness, agility, and information availability. Whether you aspire to these qualities under the umbrella of a specific management strategy or without an explicit strategy, you probably can identify where and how these qualities would provide benefit to your company.

This book provides real-world examples of how event processing can improve your timeliness, agility, and information availability, but it won't attempt to identify every possible application. Instead of trying to convince you that these system requirements are desirable, this book concentrates on explaining *how* to achieve these aspirations through the appropriate use of event processing (see Figure 1-1, Architectural Style column).

---

*Note: Here, and in many other places in this book, the phrase event processing refers to the design discipline that implements the principles of event processing, not the mere act of responding to a business event.*

The next three sections clarify the terms *timeliness*, *agility*, and *information availability*.

## Timeliness

Two types of timeliness (or "celerity") are identified here: timeliness that is exhibited by a "low latency" response to a particular input, and timeliness in the form of completing an end-to-end business process in a shorter elapsed time.

## Reducing Latency for an Individual Activity

*Latency* is the time it takes for a system to respond to an input. Latency in the user interface is one of the most critical factors in the design of an interactive system. Software engineers pay an enormous amount of attention to latency because of the impact that it has on human productivity. When online application systems first became popular in the 1970s, studies showed that applications that returned results to the display terminal within 400 milliseconds (0.4 second) after the user pressed ENTER on the keyboard provided excellent user efficiency for transactional applications. More-common systems with 1- to 2-second latencies were a bit less valuable because people noticed the pause and their rhythm was disrupted in subtle ways. Systems with higher latencies were not only annoying, they damaged the productivity and effectiveness of the business.

Latency is still an important issue for surfing the Web and for social applications. Modern web designers know that if their site can't display a new web page within several seconds of a person clicking a hyperlink, many users will lose patience and abandon the thread they are pursuing (although their willingness to tolerate delay varies considerably, depending on the nature of the task—sometimes 2 seconds is too long!).

Latency is also an issue in other aspects of IT systems and in business activities that don't use computers at all. Reducing latency can help a person or company in myriad ways, most of which are fairly obvious. For most business activities, faster is better up to a point, but beyond that point timeliness has little or no additional value. Event processing, and particularly event-driven architecture (EDA), helps reduce latency, as we'll explain shortly.

## Reducing Business Process Elapsed Time

An insurance company that used to pay 90 percent of automobile damage claims within 25 working days now pays 90 percent of those claims within 9 days. A computer vendor that previously filled orders for PCs within two weeks now ships PCs to customers within 24 hours. Banks that traditionally settled foreign currency trades overnight now settle some currency trades within seconds using real-time gross settlement (RTGS) systems.

These are examples of *multistep* business processes—sequences of activities carried out by one or more people, application systems, or business units. The benefit of reducing the end-to-end elapsed time for the process varies. In some cases, the benefit is increased customer satisfaction, leading to more revenue. In other cases, the company saves money by being able to hold less inventory, thereby reducing the carrying costs of inventory. In the case of RTGS, banks are reducing their risk.

If you consider process elapsed time closely, it turns out to be another way of looking at latency. The difference between a multistep process and an individual activity is in the eye of the beholder. Even the simplest activity can be considered a process because it can be decomposed into subactivities. For example, when you click a hyperlink to navigate the Web, you are initiating a short process that will end when the new page is displayed on your browser. To decrease the latency of an individual activity, you need to reduce the elapsed time of the process that constitutes that activity.

Latency is more useful than process elapsed time as a way to think about timeliness where you don't want to deal with the subactivities. On the other hand, process elapsed time is more appropriate when you have reasons to be aware of the component activities. People often think of latency for responses that occur within seconds or minutes and use elapsed time for things that consume hours or days, but the distinction is arbitrary and technically fictitious. Ultimately, the business only cares about minimizing the time between the beginning and the end of the activity or process, and EDA helps this in a wide range of business situations.

## Agility

Enterprise agility is one of the most trite goals in business today, but it's trite for good reason. The world is changing faster than ever. An agile enterprise can readily adjust its behavior in response to environmental or internal changes—such as shifts in customer demand, competitors' activities, regulatory requirements, economic trends, supplier activities, and company circumstances. Agility is different from timeliness because it refers to a company's ability to change its activities rather than its ability to just perform them in less time.

Here we identify two forms of agility: instance agility and process agility.

### Instance Agility

*Instance agility* is the ability of an entity to handle each instance (iteration) of a business process differently. For example, each car that comes off an assembly line can have its own unique combination of colors, tires, and audio equipment. In the extreme, no two cars may be alike. Another example is an insurance company cutting and pasting a different set of amendments (“endorsements”) to create a custom insurance policy that is tailored to a customer's particular needs. Processes that allow decisions to be made by a person dynamically at run time exhibit instance agility. Activities can be skipped, inserted, executed out of order, or modified on the fly. Instance agility enables “mass customization,” and when applied to marketing, people speak of it as “precision marketing” or selling to a “market of one.”

### Process Agility

*Process agility* is the ability of an entity to change a whole process to support new kinds of products or services. For example, if customer demand for sport utility vehicles (SUVs) drops, an automobile manufacturer may decide to convert an SUV assembly line to produce smaller, fuel-efficient cars. Another example is an insurance company that enters a new market to insure boats by adapting the process, forms, people, and application systems that it uses to insure cars.

The effect of process agility is more durable than the effect of instance agility. Process changes affect every instance that is undertaken in the revised process. Every vehicle that comes off the modified assembly line will be a car rather than an SUV. By

contrast, instance agility doesn't change the process definition; it merely provides ad hoc variability for each instance within the bounds of a defined process.

In subsequent chapters, we explain how events play a role in enabling both kinds of agility.

## Information Availability

This book addresses only a small part of the information management discipline, specifically focusing on three aspects that are most directly affected by event processing: data consistency, information dissemination, and situation awareness.

### Data Consistency

The goal of *data consistency* initiatives is to get multiple business units and their respective application systems and databases to agree on the facts. For a variety of reasons, all companies inevitably have redundant versions of some data regarding customers, products, orders, employees, and other entities. We can bemoan the disadvantages of maintaining redundant data and strive to reduce it through strategies such as master data management (MDM), but maintaining redundant data sometimes is necessary and often is inescapable in practical terms. This gives rise to the perennial problem of keeping data consistent (“in sync”) across the disparate data stores. When a person, department, or application system receives or generates new data or a revised version of the data, the person or entity updates the local database. The person or system should also forward a copy of the new data to other departments and systems that are keeping track of that data.

Business-to-business (B2B) relationships also depend on data consistency. For example, a company may need to send its product catalog and price information to its distributors or customers. This may entail bulk transfers or partial updates.

### Information Dissemination

The need to pass on information is one of the most common situations in business and in life. A person or system detects that something happened and sends a report to one or more people or systems. Notifying a single recipient is a simple case of information dissemination. Some messages report routine events that are expected in the normal course of business, such as the arrival of a shipment. Other notifications report exceptions or abnormal situations, such as when a warning system notifies all the students on a campus through e-mail, Short Message Service (SMS) text messages, or a “reverse 911” network that the school will close early because of a snowstorm. A notification that is intended to cause a response is often called an *alert*.

### Situation Awareness

*Situation* (or *situational*) *awareness* means knowing what is going on so that you can decide what to do. The term originated in military applications to describe the goals

of advanced command and control systems, but it is relevant in many business scenarios as well. Companies seek to cut the fog of commerce just as generals seek to cut the fog of war. Almost every company has operational activities that run continuously and must respond quickly to changing conditions. Situation awareness implies that you have an up-to-the-minute understanding of the environment and your own internal circumstances. The purpose of situation awareness is to help you make faster and better decisions—it can be viewed as the ultimate goal of intelligent decision management initiatives. Situation awareness goes beyond mere dissemination because it implies that someone or something is able to synthesize multiple sources of input data to develop a holistic picture of what is happening.

## Relating Business Drivers to System Design

Timeliness, agility, and information availability can't be achieved simply by speeding up traditional business processes or exhorting people to work harder and smarter with conventional applications. These goals generally require fundamental changes in the architecture of business processes and the application systems that support them. Often this involves more use of the event processing discipline.

Almost every step in every business process today is enabled in some direct or indirect manner by an IT system. You can't have timeliness, agility, and information availability in your business unless you also have timeliness, agility, and information availability in your IT systems. Of course, IT is only part of the solution. Improving the business also requires changes to the non-IT aspects of operations, including people's job descriptions, document and form design, the flow of work, corporate policies and procedures, the organizational chart, and corporate governance. These are all part of business *systems* in the larger sense of the word.

## Summary

Events—things that happen—are a central fact of life for companies and people. Companies can improve the timeliness, agility, and information availability of their operations if they handle events in a systematic way that leverages advances in the contemporary understanding of how events work.