IBM

Empowering the business to sense and respond:
Delivering Business Event Processing with IBM WebSphere Business Events

Contents

Introduction

In today's hypercompetitive global market, corporate agility is no longer a competitive advantage—it is now a matter of survival. Agility is the organization's ability to sense and respond in a timely manner to changing requirements for actionable situations that represent opportunity or risk. Complicating factors include the speed at which business is conducted and the requirement to support the execution of dynamic business processes that require a holistic view of activities occurring across the extended enterprise.

Attaining corporate agility requires the IT organization to align the computing infrastructure with the business while supporting an unprecedented level of performance, flexibility and insight—understanding what's happening, when to respond and what to do.

Supporting agility through alignment has led many IT organizations to rapidly adopt a service-oriented architecture (SOA). An SOA is a comprehensive strategy for constructing a highly responsive IT infrastructure by leveraging existing investments in applications and system technologies.

This paper addresses one of the critical IBM technologies for supporting agility and alignment, Business Event Processing (BEP). But before a deeper dive on BEP, let's take a brief tour of its technical predecessors.

Event processing

Event processing is the ability to detect and respond to events (or activities) occurring across the enterprise. An event is any electronic signal, or message, indicating a change in the state of the enterprise. For example, an event message could indicate the addition of a new customer, the sale of a product, receipt of a shipment, an open security door or the current location of an asset using GPS. Event processing has evolved from addressing simple events and complex events to focusing on business events.

Simple event processing

Simple event processing is not a new concept. For the last 40 years organizations have been using a form of simple event processing to detect and respond to a single source, or homogeneous event type. One such example is: if Event A occurs then do X. A simple event process might specify: if a "shipment received" event occurs, add the quantity to an inventory database.

Complex event processing

In large organizations, tens of millions of events occur every day, but not all events or event occurrences are of equal importance. Providing insight requires the ability to determine when a pattern of related or seemingly unrelated events from one or more sources has occurred and then to coordinate the execution of the responses to that pattern of events. Recognizing the complexity, excessive time and cost that would be required to write custom code for such a solution, a software technology emerged specifically designed to address these requirements: Complex Event Processing (CEP). See Figure 1.
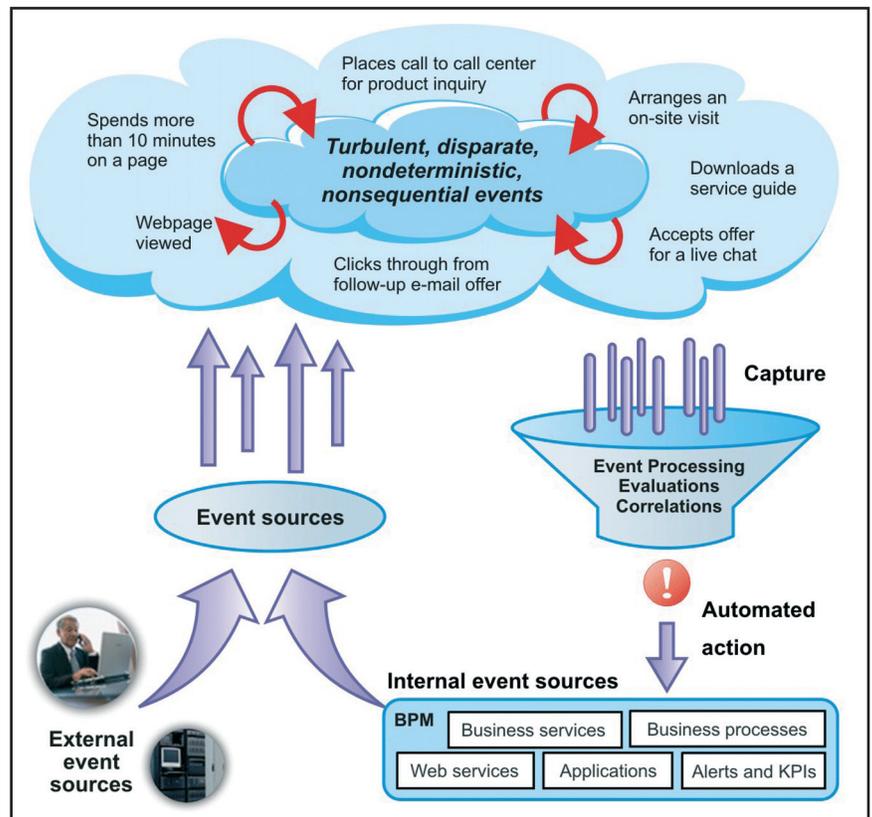

Figure 1. Overview of Complex Event Processing

CEP provides functions to help detect and respond to event patterns among like or related events, missing events and aggregate events. It also provides the ability to relate the pattern detection to a context and apply a dimension of time to the pattern, such as *if event A and B occurs and event C does not occur within <time frame>, then do X, Y and Z after <time frame>*. Here is an example of using CEP to identify the opportunity to up-sell to a highly qualified existing customer and initiate follow-up activity from the sales team:

If:
within a 72-hour period, a customer researches a specific product at the Web site and calls the help desk for more information, but does not purchase the product

Then:
send a sales follow-up transaction updating the customer/product activity in the CRM system, add the customer to the direct mail system and, after five days, send a "thank you for your interest" e-mail to the customer.

In addition, CEP supports the detection of multiple or aggregated patterns of events. For instance, detecting a "failed log-in" event pattern at a Web site might be defined as three incorrect password attempts within two minutes, whereas, a "deny access" event pattern might be defined as two occurrences of the failed log-in event pattern.

Over the last few years, a number of vendors have announced products with varying degrees of support for CEP function. Due to the architecture and feature set employed within these initial products, CEP has been most often used for specialized application requirements, including:

- *High volume of homogeneous event types*
- *Predictable time and order of events and complex patterns*
- *Event patterns that occur over extremely short periods of time*
- *Processing logic that is mostly static and maintained by IT*

As a result, CEP deployments are currently found primarily within the financial services sector, supporting applications such as stock trading. There, buy and sell order events typically occur within seconds, within a predictable sequence, and the processing logic changes infrequently.

Business Event Processing

To address the event processing requirements across industries and application domains and to improve consumability within the enterprise, additional capabilities are needed. This need has given rise to the development of a new CEP-based technology representing a convergence of power and ease of use in support of corporate agility: Business Event Processing (BEP). See Figure 2.

BEP extends CEP capabilities and provides a graphical, non-programmatic user interface that allows business users to manage event processing logic themselves, resulting in a significant reduction in time to value and total cost of ownership.

BEP is distinguished from CEP in its ability to support a broader set of application requirements, including:

- *High volume of heterogeneous business event types from multiple sources*
- *Business events and complex patterns that occur in no particular time or order*
- *Frequent changes to event processing logic*
- *Event processing logic maintained by business users themselves*

These advanced features and the incorporation of a graphical interface for business users results in an increase in flexibility and reduction in time to value.
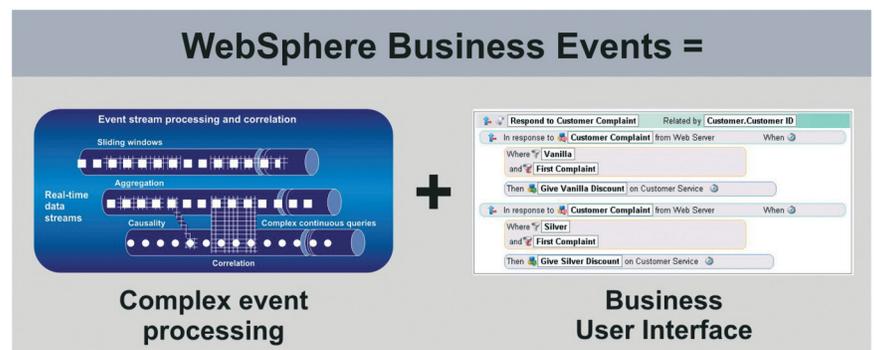

Figure 2. Overview of business event processing

BEP supports event processing requirements, and the need for increased responsiveness to change, across industries and application domains. Here are examples of applications that benefit from the capabilities supported by BEP:

- *Application review and approval*

- *Case management*

- *Compliance*

- *Customer service*

- *Exception detection (alerts)*

- *Fraud detection*

- *Identity theft*

- *Marketing and sales orchestration*

- *Supply Chain Optimization*

- **And MORE!**

WebSphere Business Events

IBM WebSphere® Business Events is an IBM software product specifically designed to support Business Event Processing by meeting the high-volume demands and processing required across industries and application domains. Equally important is enterprise consumability—the extensive use of graphical, codeless user interfaces in WebSphere Business Events greatly simplifies implementation and empowers business users to directly develop and maintain event processing logic.

WebSphere Business Events consists of these basic constructs:

- *Connectivity to business events*
- *Event processing engine for evaluating and correlating business events*
- *Initiation of business responses (actions)*

Business events can exist anywhere within the extended computing infrastructure, both inside and outside the firewall. Events can be communicated directly between systems or pushed into the communications backbone for use by any system (such as through publish/subscribe or request/reply).

Business events sometimes contain associated data (but not always), and they can occur or not occur across the extended enterprise. Business events include real events, synthetic (derived) events and event patterns:

- **Real (raw) events** *are created by human interaction, a system, or a device. They indicate that a certain activity has just occurred. For example, an event might indicate that a prospective client has requested information on the Web site, a specific product has arrived at the warehouse, a change in supplier price has occurred, the temperature has changed, or an assembly process has stopped.*
- **Synthetic (derived) events** *are created dynamically and are reprocessed by the event processing engine. Synthetic events support the requirements for recursive processing, which is useful for aggregating event data or simply indicating that another instance of a certain event has occurred. This aggregation, in turn, supports the detection of a certain complex pattern of related or seemingly unrelated events. A synthetic event might be generated to maintain the average temperature over the past three days, document the lowest sale price in the past 30 days, or trigger delayed-event processing logic to determine whether an event pattern has occurred (or not occurred) within a specified time frame.*
- **An event pattern** *is one or more occurrences of the same or related events (real or synthetic). User-defined events that form a pattern can occur at any time or in any order. Here are some examples: an online user reaches the maximum of three failed log-in attempts on two different occasions, or a prospective client requests a price quote from the Web and the help desk on two different occasions but has not purchased the products within seven days of the most recent request.*

The *processing engine for business events* is a software system designed for detecting, evaluating, correlating and responding to certain event occurrences and patterns. The processing engine receives or requests certain events, runs user-defined processing logic, and—when defined events or patterns are detected — creates and communicates action messages.

A *business action* is an electronic signal or message created by the processing engine and communicated directly to systems (or over the communications backbone) to indicate that an actionable event or pattern has been detected. Action messages sometimes contain associated data, but not always. Actions can be thought of as result-events.

WebSphere Business Events components

WebSphere Business Events includes design and runtime environments. These environments are integrated through a shared object repository. See Figure 3.

WebSphere Business Events

Design Environment

WebSphere Business Events

Object Repository

Evaluations
Correlations

WebSphere Business Events

Runtime Environment
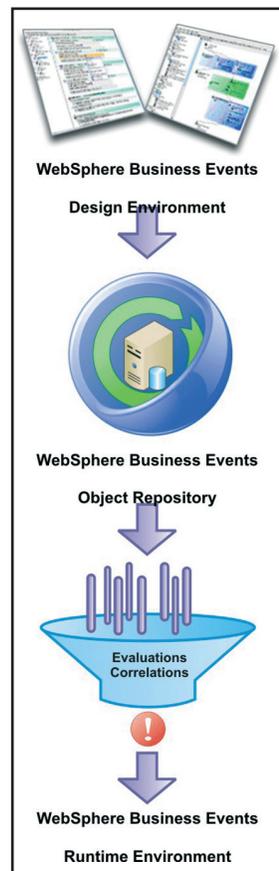
Figure 3. Components of WebSphere Business Events

The design environment

The design environment supports a collaborative process for both IT and business users to define and manage the event processing system. The design environment provides a graphical user interface for defining all of the assets required for business event processing.

Recognizing their responsibility to manage and control the information infrastructure, IT defines the Building Blocks: information describing events, actions, certain conditions (filters), and user console forms that will be made available for processing. Furthermore, recognizing the business users' inherent knowledge of what events and patterns represent actionable situations and what corresponding actions should be initiated, they are able to define and manage the processing logic and dashboard graphs and charts themselves.

<u>Defining the building blocks (events and actions)</u>

Using a GUI, IT defines the building blocks for Business Event Processing: the events and actions, their payloads of data, and the connection strategies for receiving and delivering them. After these building blocks are defined, they are made available through an abstraction layer that insulates the person specifying the event processing logic from the low-level details and definitions. This abstraction layer enables these elements to be exposed in business terms. See Figure 4.
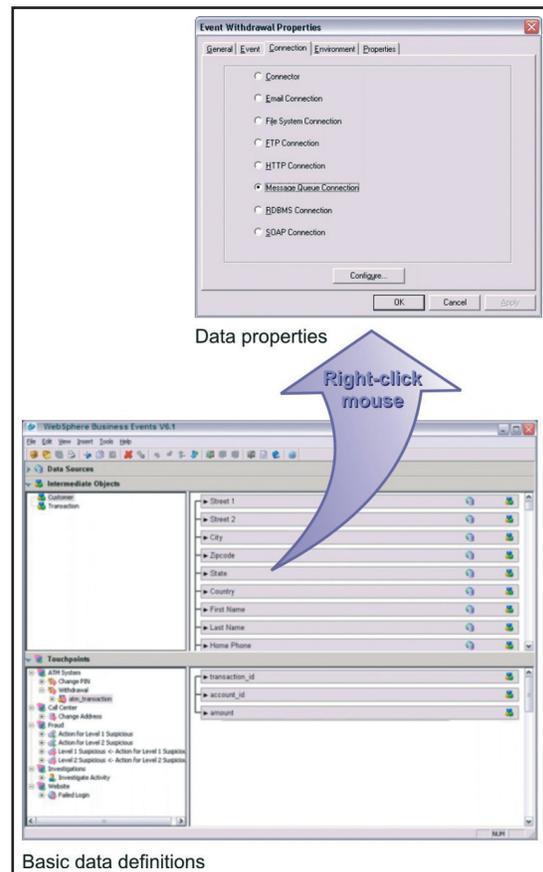
Figure 4. Defining event and action building blocks through data definitions and properties

Defining business event interactions (logic)
WebSphere Business Events enables both business and IT users to specify the event interactions (processing logic) by providing a graphical, non-programmatic interface, eliminating the typical need to use either typical Event Processing Languages (EPLs), which are most often exposed as a form of extended SQL programming language, or scripting language interfaces, which require programmers.

The event and action building blocks previously defined by IT are used to populate a drop-down list from which business users can simply point and click when constructing event interactions. To ensure the integrity of the processing logic, the user interface exposes only logical and permissible events, conditions and actions for selection.

Defining conditions (filters) is part of defining event interactions

Conditions can be used to evaluate data and/or define event patterns (for example, you might define a Platinum Customer condition as a customer with a level of purchases greater than US$225,000) or you might define the pattern for a "failed log-in" condition as a "failed log-in" event occurring three times. All conditions are reusable and available for selection when defining event interactions.

Defining event interactions is a matter of selecting pairings of events, conditions and actions using Point/Click from drop-down lists built by WBE from the information provided as part of the Building Blocks defined previously.

Figure 5. Defining event interactions

Defining charts and graphs for the dashboard

The WebSphere Business Events dashboard can display charts and graphs to illustrate any activity processed by the event processing engine. The Web-based interface is designed to support business and IT users. It uses point-and-click and drag-and-drop interactions, and greatly simplifies the creation of charts and graphs by exposing only relevant and permissible assets for selection. See Figure 6.

Figure 6. Examples of charts displayed on the WebSphere Business Events dashboard

Defining tasks and alerts for the user console

The WebSphere Business Events user console is a Web-based form, customized by IT, that displays task lists or alerts resulting from the processing of events. See Figure 7.
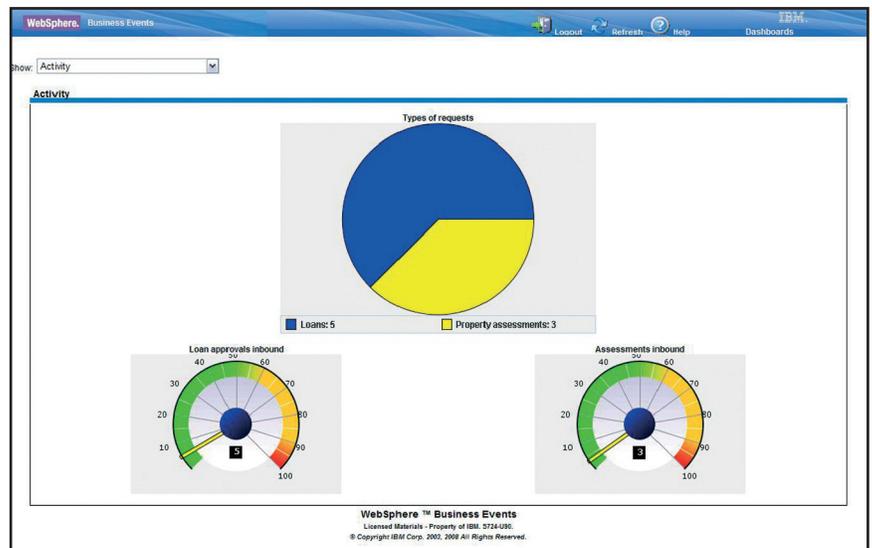

Figure 7. Example of information displayed on the WebSphere Business Events user console

The object repository

The object repository provides a means of maintaining the asset information that controls the event processing environment, as well as supporting incremental life-cycle development and deployment, and access privileges. (See Figure 8.) All defined assets in the repository are maintained as decoupled objects, which supports reuse and incremental deployment. Definitions of events, actions and data, connectivity, interactions, and conditions (filters) are all decoupled at design time. The repository is also a powerful semantic layer of abstraction that is critical for empowering business users to interact with the systems, by allowing them to use their business vocabulary to manage the processing logic themselves.
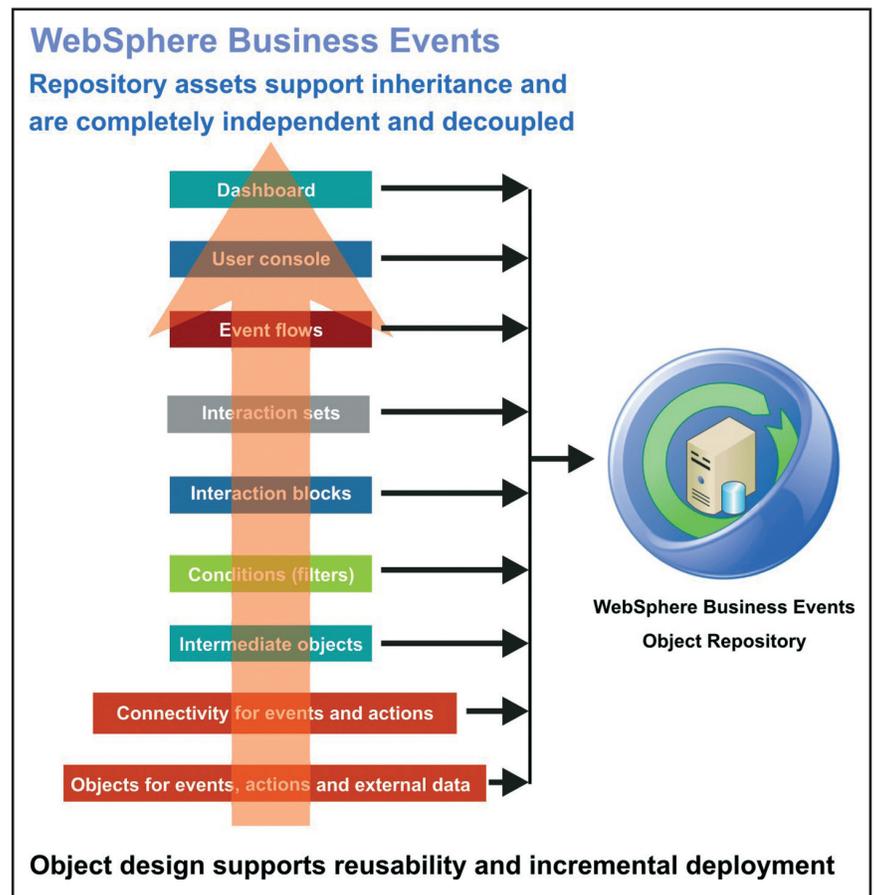

Figure 8. Contents of the WebSphere Business Events object repository

The runtime environment

The driver of the WebSphere Business Events runtime environment is the processing engine for business events. This processing engine is an asynchronous network application. When initiated or refreshed, the processing engine loads the object repository into memory. The user-defined event interactions define the processing logic to be executed. When defined event interactions (actionable activities) are detected, the engine coordinates follow-on processing by creating and communicating actions. See Figure 9.
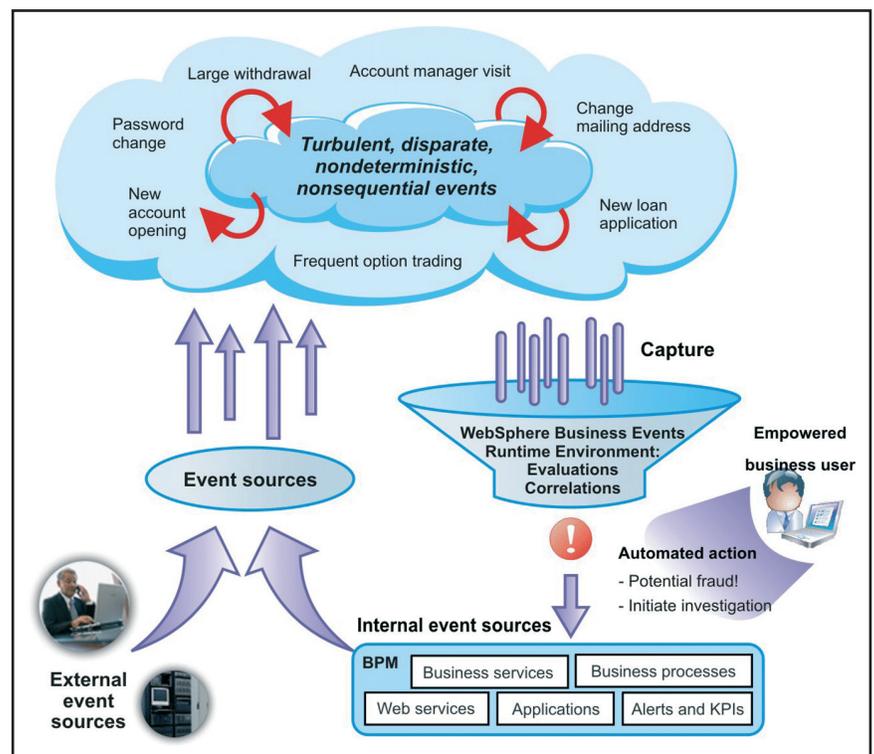

Figure 9. The WebSphere Business Events runtime environment

Synergy with BPM and BAM

Distinct methodologies for business process management (BPM), business event processing (BEP) and business activity monitoring (BAM) are designed to address different problems and are beneficial as stand-alone implementations. However, there is a tremendous amount of synergy when

these methodologies are integrated. BPM deals with the definition, modeling, simulation and execution of business processes and includes other methodologies such as Business Process Execution (BPE) platforms. So, while BEP enables detection of existing, actionable business situations and provides coordinated responses, BAM is used to monitor KPIs, discover exceptions and provide visibility through charts and graphics. When integrated bidirectionally, these methodologies provide both notification and response to what is known by the other.

For example, a BEP product such as WebSphere Business Events might notify both a BPE platform and a BAM utility that it has discovered a critical event pattern. In response, the BPE platform might execute a new, synchronous business process while the BAM tool updates a graph and highlights any exception situations. Upon completion, the new business process might communicate its result to WebSphere Business Events (as a new event) and the BAM utility might communicate an event indicating that an exception condition has been detected. This closed loop provides the ability to know what's happening, when to act and what to do. The combination achieves a new level of dynamic business process execution. See Figure 10.
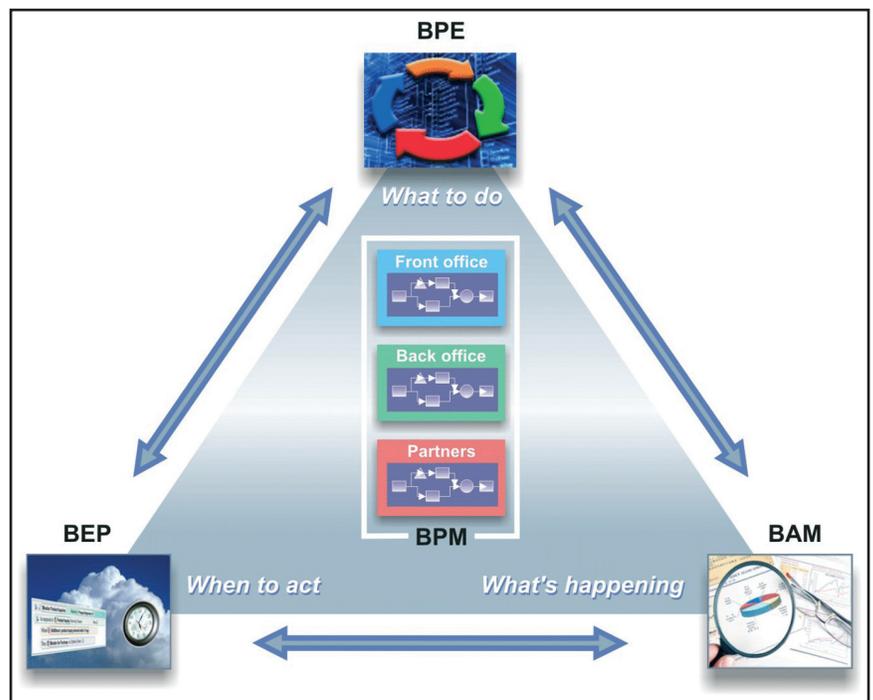

Figure 10. The synergy between business process management, business event processing and business activity monitoring

Synergy with SOA

Another example of the synergy afforded by BEP is its interaction with a service-oriented architecture (SOA). SOA is an architecture that provides an enterprise with a service-oriented view of their business processes and aligns the IT components with the business components.

BEP increases the agility of an SOA in several ways:

- *The applications enabled by BEP increase the control of business users over their own business: it enables observation of significant phenomena soon after they occur and helps identify risks, opportunities and other business situations that require a response.*
- *After proper setup, BEP provides the ability to transfer information among different services and applications using events in a decoupled way. As a result, consumers and producers of this information no longer need to understand the underlying internal programming.*
- *BEP allows business users to define, modify and understand the behavior and functions of the Business Event Processing system. This support decreases their dependency on IT resources and budgets and enables business users to modify process behavior, such as by adding or modifying patterns or changing routing policies.*

BEP also works with the SOA reference model. Services can emit events and consume the results of the event processing. The event processing itself is performed as an extended capability of the enterprise service bus (ESB), with parts that can be closer to the consumer or producer of the events. BEP can trigger a business process that is composed of a workflow whose state is sensed by the BEP system, which in turn sends a notification or orchestrates further actions. See Figure 11.
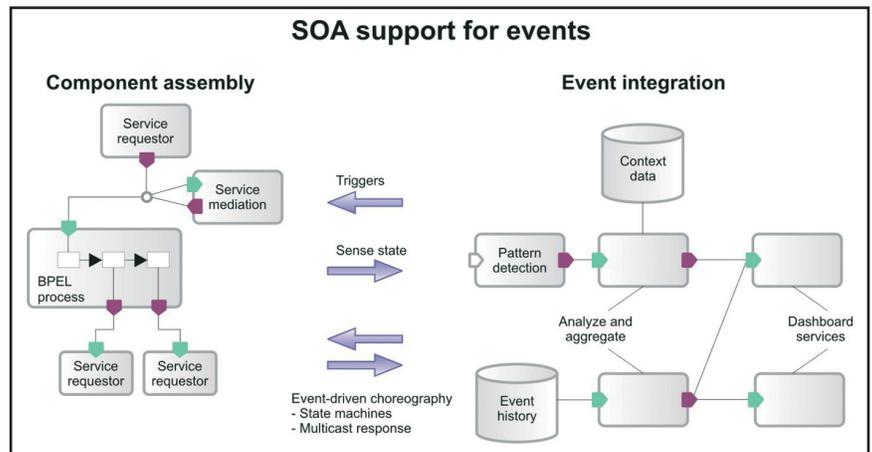
Figure 11. SOA support for events

Summary

Business Event Processing, and specifically the IBM WebSphere Business Events product, is designed to improve corporate agility and address the requirement for organizations to dramatically improve their ability to sense and respond to opportunities and risk as it occurs. Based on years of experience in event processing technology, IBM's introduction of WebSphere Business Events takes event processing to the next level of power, flexibility and time to value by empowering the business user to directly manage the event process themselves. For those businesses engaged in an SOA, the availability of WebSphere Business Events as an integrated component extends the power and flexibility for achieving this goal.

For more information

To learn more about the WebSphere Business Events product, please contact your IBM marketing representative or IBM Business Partner, or visit the following Web site: **ibm.com**/software/integration/wbe

IBM®

TAKE BACK CONTROL WITH **WebSphere**®

Recyclable, please recycle

WSW14031-USEN-00