



BPM 2.0: Process Without Programming

Vitria BusinessWare 4.1

Integrating the Extended Enterprise

Every ten years or so, the dynamics of business are transformed by the combination of new macroeconomic forces and new technology enablers, and these dynamics in turn reshape corporate investment in information technology. In the 1990s, the forces of globalization suddenly made labor productivity and operational efficiency critical corporate objectives. Transformed by new computing power at the desktop and networking throughout the company, IT investment focused on automating inefficient manual processes and integrating information within individual functional areas of the business. The result of that effort has been workflow automation of most paper-driven processes, and widespread deployment of packaged mega-applications for ERP, financial systems, customer relationship management, and human resources.

Today, a new dynamic – internet e-business – is once again transforming the goals of IT investment. Companies are now racing to become digitally connected to their customers, suppliers, and partners. Instead of efficient mass production they now want efficient “mass customization,” the ability to tailor products and services to the precise needs and preferences of individual customers without sacrificing productivity. This in turn means businesses face mounting pressure to focus internal resources on their core strengths and rely on partners to provide the rest of a “whole product” offering. Behind a company’s *brand* – the facade it promotes to customers – increasingly stands an *extended enterprise*, a dynamic value chain of suppliers and business partners interconnected over the internet. Today the overarching goal of IT investment is no longer simply lowering production costs, but rather increasing the company’s attractiveness to customers and partners in an digitally connected world.

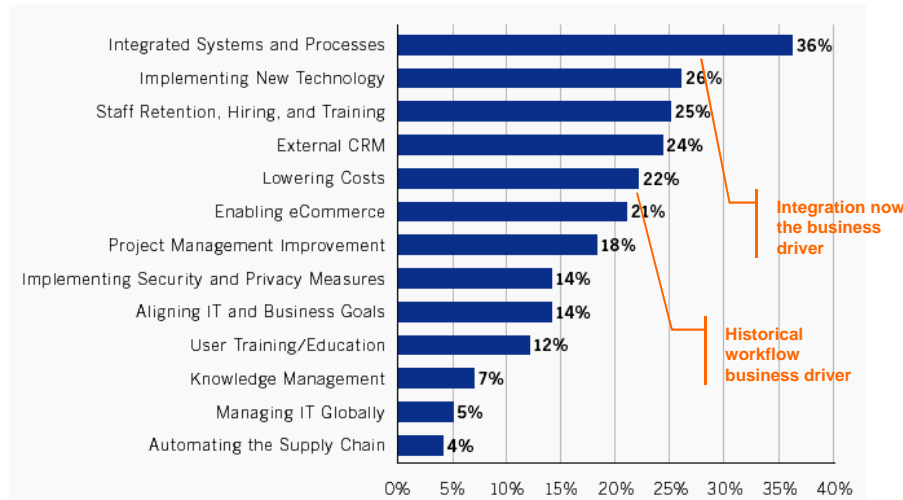
What’s New About Business Process Management?

Business process management (BPM) is the term used to describe the new generation of technology designed to automate, integrate, and optimize processes across the extended enterprise. While it shares some characteristics with the workflow and business process re-engineering technology of the 1990s, BPM is based on a new set of IT objectives, new business values, and a new underlying technology infrastructure.

New IT Objectives

As in previous periods of changing business dynamics, internet e-business is focusing management attention on *business processes*, but the fundamental objectives of process improvement have changed. Where earlier generations of process management technology tried to stamp out inefficiency by automating isolated functional bottlenecks, the goal today is to interconnect the myriad islands of process automation created by that earlier effort. E-business demands integration of end-to-end processes that cross functional boundaries, even spanning the entire extended enterprise.

Process integration has thus replaced *process automation* as the critical IT objective. In a recent *CIO Magazine* survey of IT executives, integrating systems and processes was the number one priority, outranking cost reduction, CRM, and e-commerce¹. While functional automation initiatives such as ERP systems, CRM systems, HR systems, and document-processing workflow systems have produced efficient, automated process flows within each function, today's goal is bringing management control to end-to-end processes, such as from the receipt of an order to fulfillment and invoicing.



Source: CIO Magazine, March 2002

Figure 1. Integration, more than simple cost reduction, is the key IT business driver today.
Source: CIO Magazine

The cost of integration just within an enterprise has turned out to be enormous, up to 40% of IT investment. Extending that integration to trading partners adds another dimension of complexity and cost. Yet that is what the new business dynamic is demanding.

Integration implies more than mere automation across diverse IT systems. Managers want to be able to *manage* processes more effectively, monitoring in real time business-oriented key performance indicators based on aggregated information from disparate systems across the enterprise, and reconfiguring resources and process flows to optimize those indicators.

New Business Values

Like the workflow technology of a previous generation, BPM still aims to boost productivity and enforce best practices, but it elevates to equal importance a new set of business values addressing the new dynamics of e-business:

1. **Agility, or responsiveness to change.** As the e-business era unfolds, change is the only constant. New customer demands, new business models, new technology platforms, capabilities, and standards emerge almost daily. BPM aims to let companies bring new products and services to market more quickly and adapt processes more effectively to changing market demands.
2. **Flexibility, or accommodation of diversity and exceptions.** It's easy to integrate a process when you code all its pieces from scratch. But companies have millions already invested in islands of process automation, each with its own system and data architecture, as do each of their trading partners. BPM

¹ CIO Magazine, March 2002

must be able to manage end-to-end processes assembled from dissimilar systems and data models. Moreover, BPM must deal with dynamic business exceptions – an order is changed or cancelled, or only part of an order is available – in an automated, transactional manner.

3. **Visibility, or management insight into business processes as a whole.** In the era of dynamic e-business, managers want real-time information on key performance indicators at the end-to-end process level. These metrics typically require correlation of data from disparate systems across the enterprise, or across the extended enterprise. BPM needs to deliver real-time visibility of end-to-end processes through a management dashboard of aggregated process data. This function is at the heart of what analysts like the Gartner Group are calling the “enterprise nervous system... the integration infrastructure or intelligent network that continuously monitors the state of the heterogeneous enterprise and its relevant partners”².

New e-Business Infrastructure

Because it is inherently about e-business, BPM technology must be layered on the new e-business technology infrastructure, in particular Java 2 Extended Edition (J2EE) and Microsoft Windows services, XML, and web services. Any new e-business initiative in a company invariably is going to be built on these new web-centric architectures and components. At the same time, BPM must provide a coherent middleware platform for integrating existing mainframe and client/server systems throughout the enterprise, and a bridge connecting legacy applications to the new architecture.

The New Formula: BPM = (Workflow + EAI + B2Bi) x Web

BPM is more than process automation, or workflow. It adds conceptual innovations and technology from two previously distinct software categories – Enterprise Application Integration (EAI) and Business-to-Business Integration (B2Bi) – and reimplements all of it on the new e-business infrastructure based on web and XML standards. BPM software thus really means the combination of workflow, EAI, and B2Bi components in an integrated modeling and execution environment based on web technology standards.

The key concept of business process management is that a process can be completely described by a *model* that clearly separates the sequence of process activities and the information flows between them from the physical *resources* – people and systems – that implement the activities. The model provides a visual description of the process that has meaning to business analysts and managers, and at the same time represents executable code that actually automates, integrates, and monitors the process.

This concept is not new. The separation of “flow logic” from “task logic,” using a graphical process model linked to a runtime environment for automating, monitoring and optimizing the process, goes back over a decade to the workflow software of the 1990s. That earlier effort, however, was centered on the replacement of manual, paper-based processes with automated flows of interactive computer-based tasks over local area networks. BPM is centered on integration of islands of information into end-to-end flows spanning the extended enterprise.

To understand what that encompasses, let’s look at the pieces individually.

² D. McCoy, “Business Process Management: Core to the ENS,” Gartner Group Commentary, April 3, 2001.

Workflow

Many BPM vendors today use the term *workflow* to simply mean process activities involving human interaction, but historically workflow software has dealt more broadly with *process automation*, including both interactive and fully automated activities. Although workflow has matured as a standalone software category or market, its basic elements have been embedded in virtually every multi-user packaged application, from ERP to customer relationship management to supply chain management to web publishing. Moreover, its core innovations and artifacts, such as *graphical flow models*, *state management* of long-running processes, and *role-based* task distribution, remain at the heart of BPM today.

Workflow technology models and automates “long-running” processes lasting days or months. Flows include both automated tasks and human interaction, with interactive tasks distributed to users based on organizational roles. Historically, workflow has tended to relegate integration functions, such as synchronizing data between disparate packaged and legacy applications, to custom code within its activities – and thus *outside* the scope of the process model. Moreover, its tightly coupled integration style using low-level APIs has historically confined workflow to local, homogeneous system environments, such as within a department or division.

EAI

Enterprise Application Integration (EAI) technology emerged in the late 1990s to tackle a different aspect of the islands-of-information problem, automated data synchronization within the enterprise: *If a customer’s address or account status changes in the ERP system, it should be updated automatically in the CRM system*, and vice versa, to keep these information islands in sync. EAI software components such as integration brokers and connector software were developed to provide a consistent framework for detecting the changes to one application or system (*events*), transforming the data to match the structures and formats used in the second system, and communicating the transformed data to that second system.

Unlike workflow, which historically has relied on a tightly coupled integration style based on *remote procedure calls*, EAI introduced a more flexible, loosely coupled style of integration via *messaging*. Messaging allows integration of applications with dissimilar platforms and APIs, and tolerates interruptions in the communications link or unavailability of the application. EAI also involves a process flow of sorts – usually a simple, short data flow between applications – but for the most part has not tried to automate long-running end-to-end processes involving many steps, nor include interactive activities in which a person (rather than an automated program) does some work and makes a decision.

In addition to *event-triggered actions*, *data transformation*, and *messaging-based integration*, EAI introduced another critical element generally missing from workflow: *transaction-awareness*. EAI software can aggregate multiple process activities as a single *unit of work*, or *business transaction*, with rollback recovery and compensating activities explicitly defined in the model and executed automatically at runtime.

B2Bi

Internet e-business adds a third technology to the mix, Business-to-Business Integration (B2Bi), the software that integrates trading partners on the web. B2Bi is the logical successor to the proprietary, batch-oriented EDI technology of the 1990s, transformed into real-time interaction over the internet using XML and standard messaging protocols. (In reality, B2Bi today *integrates* EDI – still the dominant mode of B2B commerce – into the new framework rather than simply replacing it.) Because both EAI and B2Bi involve

“transactions,” some EAI vendors position B2Bi simply as “EAI beyond the firewall.” However, B2Bi is fundamentally different from EAI in two important respects.

First, B2Bi does not connect directly to the IT systems of trading partners, but just manages the sequence of standardized messages exchanged by the *public interfaces* of each trading partner. For example, the public interface may require that a seller must respond to a *PurchaseOrder* message with either a *POAcceptance* message or a *PODenial* message, but seller’s internal process that determines which response is chosen is outside the scope of B2Bi. It is part of the seller’s internal business process.

Second, a single process definition, such as handling a purchase order, must be reusable with any potential trading partner, each varying in communications capabilities and standard terms of business agreement. B2Bi thus has to provide a runtime environment that allows buyer and seller to dynamically negotiate *trading partner agreements (TPAs)* that specify the communications parameters and terms of agreement governing the B2B process.

Today B2Bi technology, including the messaging protocols and documents exchanged, as well as their associated TPA management framework, is increasingly being standardized in initiatives such as ebXML, RosettaNet, and web services.

Web-Centric Architecture

The prospect of conducting business with customers and partners electronically is so powerful that even in its formative stages the new e-business infrastructure is now the basis for virtually *all* new major IT development. Adapting to XML and the new J2EE and Microsoft web platforms represents a wholesale architectural change cutting across the entire enterprise IT infrastructure, not unlike the shift from proprietary mainframes and minicomputers to client/server “open systems” in the 1990s. Thus the fourth thread of BPM technology is a *platform shift* to the e-business infrastructure, in particular *web application servers* and the powerful integration potential of XML-based *web services*.

Web application server platforms already provide database connectivity, transaction management, EAI-style connectors, message queuing, and native web services support, and are evolving into business process execution engines. This invites the question driving one side of the BPM vendor competition, should BPM be part of the application server, leveraging common platform services such as security and transaction management? Or should it be a separate platform, dedicated to the specific needs of BPM, and simply provide integration hooks to app servers? It should be no surprise that app server vendors, integration server vendors, and workflow vendors have different opinions on that question.

Web services technology represents the next generation of BPM infrastructure, in which all application resources throughout the extended enterprise will be exposed as “services” invoked using XML messages in a standardized format (WSDL), delivered over standard communications channels using a standardized protocol (SOAP). In fact, processes themselves will be exposed externally as web services. Even before web services standards evolve to provide uniform mechanisms for required BPM functionality, leading BPM vendors will layer their own value-added implementation of this functionality on top of existing core web services standards.

Fundamental New Capabilities

All of these independent threads – workflow, EAI, and B2Bi – are coming together as the technology foundation of BPM. But the whole is more than the sum of the parts. BPM is about some fundamental new concepts:

- **Managing processes end-to-end.** Historically each component technology has just managed a *piece* of the business process, not an entire customer- or partner-facing process from beginning to end, such as order-to-ship, including both application-to-application integration and human interaction. The ability to combine process fragments separately controlled by EAI, workflow, and B2Bi components into a single end-to-end process model promises a new level of management control and business activity monitoring. It also allows enterprises to leverage the services of extended enterprise partners through standard public interfaces, without losing end-to-end visibility.
- **Driving process implementation from business-oriented models.** In the past, business analyst-oriented process models have been used mainly as requirements-gathering tools, providing guidelines for separate implementation models built by developers using different tools. It usually doesn't take long for the business model and the actual implementation model to get out of sync. BPM promises to put businesses back in control of their processes by generating implementation models and code directly from the business-oriented models.
- **Monitoring process activity in real time and dynamically optimizing via business rules.** This is what vendors mean when they say business process *management* is more than just process *automation* or process *integration*. Managing processes first requires aggregating process data in business-oriented metrics such as key performance indicators and balanced scorecards. If the process is “out of bounds” or service level agreements are not being met, the next step is to optimize it by reconfiguring resources or modifying business rules – dynamically, and “on the fly.”

It sounds great, but can BPM software really do all that? From the vendor web sites and brochures, the answer is a resounding YES. A more accurate answer is... sort of. Some of it today, some in the proverbial “next release.” It's all on the “roadmap.” It just depends how much custom code you want to write.

Three Promises of BPM

BPM means more than simply end-to-end process automation. BPM suppliers are making three basic promises to the business executive, dealing with process *modeling*, *integration*, and *management*.

The Modeling Promise

In BPM, process implementations are driven directly by *process models*. A process model is a *template* for the process, a detailed description of the sequence of activities involved under all possible conditions, including exceptions, and the information flows and transformations required to support each activity. Process models are usually created in the form of *graphical flow diagrams* in which nodes of various shapes represent different types of process activities, and the arrows that connect the nodes represent flows of control or data. Such visual representations can be drawn by non-programmers and, in theory, make the process logic easy to follow. However, underlying the visual representation is a precise specification of the process, conforming to a *process modeling language*, which can be compiled into components that run on a *process engine* to actually *execute* the process flows.

BPM promises to give business analysts, not just programmers, the visual design tools needed to create true end-to-end process models combining process automation, human interaction, and application integration. ***Central to the BPM value proposition is the***

promise that business managers and analysts do not simply create “business requirements” that programmers translate into technical implementation, but can themselves define the process logic that – somehow – actually executes the process.

The Modeling Challenge

Historically, process modeling has been used to analyze and document business requirements for software developers, but not to create actual implementation code. A common problem is that the technical detail required to make a process model executable may be unknown to the business analyst, may require custom programming, and generally gets in the way of understanding the process at the business level. In practice, this means that the implementation and the model rarely stay in sync. It also means the model cannot be used at runtime to monitor business activity, nor be optimized dynamically based on management metrics. ***Thus the modeling challenge for BPM is to create a visual paradigm for allowing non-programmers to specify all the parameters and details required to make process models executable.***

The Integration Promise

In BPM, the process engine does not actually “do the work” represented by the business process. It merely directs the flow of information and activity requests to external resources, such as applications, databases, or other processes, at the proper time and in the proper format, and returns response information from those resources. Those resources actually do the heavy lifting. A fundamental assumption of BPM is that the resources that implement process activities do not conform to any homogeneous system architecture but rather represent a diverse set of platforms, programming languages, and data structures. Most of them, including packaged and legacy applications, already exist in the organization. They are not created as part of the process model, and often cannot even be modified. Nevertheless, BPM must be able to integrate information from all these diverse application systems in order to execute and manage the process.

A second fundamental promise of BPM is a framework for integrating the process model with IT resources including application systems, information stores, and external processes despite differences in platform architecture, physical location, programming language, and data models – even when the source code of those resources cannot be modified.

Moreover, BPM is *transactional*. In a transaction, multiple process activities treated are as a single *unit of work*, so that failure of one activity voids the completion of other activities in the transaction. When the activities can take place nearly simultaneously and their resources can be coordinated – such as debiting an account in one database and crediting an account in another – standard transaction protocols such as *two-phase commit* can maintain data integrity by rollback to an initial state. However, in long-running processes, the activities making up a *business transaction* may be widely separated in time or managed by different business entities, requiring new transaction recovery models including compensating activities to undo the effects of partially completed transactions.

BPM thus further promises the ability to aggregate process activities into transactional units through an extended transaction model framework for committal, rollback, and compensating activities, to maintain process integrity despite the heterogeneous nature of process resources.

The Integration Challenge

EAI provides solutions to these integration challenges, but not in the context of complex, long-running processes. Workflow has a long history of automating complex, long-running processes, but has usually left both application integration and transaction management up to the resources themselves rather than providing a consistent framework within the process model. ***The integration challenge is to marry EAI's loosely-coupled, event-driven, transactional integration paradigm with workflow's state management of complex, long-running processes – while retaining scalability and performance.***

The Management Promise

BPM vendors stress that their software provides more than process automation and integration, but process *management*. What they mean is global visibility of running processes. ***BPM promises the ability to monitor both the state of any single process instance and all instances in the aggregate, using present real-time metrics that translate actual process activity into key performance indicators (KPIs).*** Management visibility is expressed in the form of real-time and historical reports, and in triggered actions. For example, deviations from KPI target values, such as the percent of requests fulfilled within the limits specified by a service level agreement, might trigger an alert and an escalation procedure.

The Management Challenge

Historically, neither workflow nor EAI technology has solved this problem. EAI has lacked the context of a long-running process in which to correlate events or transactions. Workflow has lacked a framework for combining data from its own activity logs with external data sources required to calculate business-relevant KPIs. ***The management challenge, therefore, is to create a unified system for storing and managing process automation (workflow) events, integration events, and external data that can be queried to create the necessary KPIs, reports, and alerts.***

BPM Functional Requirements

Baseline Requirements

To flesh out these implicit promises, over the past two years industry analysts have described a consistent vision³ of what BPM technology should provide – and, by implication, *actually does provide*. In essence, their functional “pillars” define what makes BPM different from its precursor technologies. Many of these functional requirements seem obvious, but actually represent capabilities far beyond those of previous generations of workflow and EAI technology. In some cases they go beyond what today’s BPM offerings can deliver as well, even though the core set of capabilities is claimed, explicitly or implicitly, by the websites and marketing collateral of virtually all BPM software providers.

The following set of core functional requirements is my restatement of the industry analysts’ “BPM Pillars”:

1. Model-driven execution

This sounds almost trivial, but it’s truly fundamental. A *process model*, which is an abstract description of the process in terms of activities and flows, automatically generates the runtime components required to execute the process. In the past, process models have

³ A good example is T. McDaniel (Hurwitz Group), “Ten Pillars of BPM,” eAI Journal, November 2001.

commonly been used to *document* an existing or proposed process, *analyze* and *optimize* it in terms of cost or complexity, or *specify functional requirements* for a process implementation built by others. In fact, products that create models that do only that, from vendors like IDS Scheer, Casewise, Rational, and Popkin, remain important software engineering tools today. But BPM models are different. They are *executable*. The descriptive model contains sufficient detail that it can be loaded onto a process engine (or compiled into runtime objects that are loaded onto the engine) and *automate* the process it describes.

2. Externalized process logic

Again, this sounds obvious, but it represents a big change from current practice. The process model should completely describe a business process in terms of what happens and in what order. Resources implementing individual process activities, such as enterprise applications or workflow tasks, implement the actions in the model's prescribed flow. However, in many enterprise applications like CRM or ERP, a sizeable amount of process logic is not exposed externally but is embedded in the application. When these applications are invoked as process activities, their embedded process logic is invisible to the process model. It is not modifiable by a process designer, and the related processes invoked by the applications cannot be correlated with the process model.

BPM thus demands what Gartner calls a new "culture of the explicit process,"⁴ in which embedded processes must be discovered, analyzed, and made explicit as a software layer controlling data flow and sequential task execution by multiple applications. Externalizing process logic in BPM makes it possible, for example, to change or upgrade a resource implementing a specific process activity without changing the model, or conversely, to alter the end-to-end flow without modifying participating applications.

3. Graphical modeling tools

Executable models are complex computer programs that automate the sequence of activities in a process, track its state, and integrate the data flows between the process and all resources. Nevertheless, modeling tools are assumed to be graphical, representing "visual programming." Process flow is modeled by assembling shapes representing different activity types, and data transformation is similarly modeled by graphically mapping data elements of source and target systems. While the model is created graphically, it is saved in a process modeling language format, which is used to generate the runtime components.

4. Support for all process and activity types

This means a process model can combine so-called "straight-through" application-to-application flows with interactive workflow tasks, including long- and short-running processes, public and private processes, and synchronous and asynchronous interactions over various communications protocols. It is shorthand for saying that BPM combines workflow, EAI, and B2Bi capabilities in an integrated software suite or package.

5. Nested and Chained Subprocesses with End-to-End State Management

There is no precise definition of what *end-to-end* means, but usually it begins with a trading partner request for a product or service and concludes with the fulfillment of that request. The usual example is *order-to-ship*, beginning with receipt of a purchase order and ending with delivery to the customer and invoicing. In property and casualty insurance an end-to-end process might start with a request for quote in the call center and end with delivery of policy documents to the customer. End-to-end processes are by nature *long-running*, taking

⁴ D. McCoy, "The Culture of the Explicit Process," Gartner Group Commentary, April 2001

weeks or even years to complete, and typically cross organizational, application, and system boundaries.

End-to-end processes are typically composed of component subprocesses representing different subsystems or organizations within the enterprise. These subprocesses may be *chained* end-to-end or *nested* hierarchically within a parent process. Support for chained and nested subprocesses has several interpretations. Ideally, a BPM offering should support all of them. The first is simply that process diagrams can provide a high-level view of the end-to-end process in which the relationships and connections between the component subprocesses is clearly described. Each nested or chained subprocess can then be exploded in a detailed diagram that shows its component activities and flows. The second is that different users or organizations within the enterprise can model their respective subprocesses independently, yet all can be integrated in an executable end-to-end process.

The first two interpretations have more to do with the notion of subprocesses than of nesting and chaining. The most important technical implication of nesting and chaining has to do with how these subprocesses are linked together in a shared process context for exception handling. For example, if the process as a whole is canceled by an external event, how is this information communicated to all the subprocesses so that the state of the process is consistent across all its component parts? Support for nested and chained subprocesses really implies a mechanism for communicating exceptions and coordinating state and exception handling across subprocesses within the end-to-end process. Only a few BPM products really do this well.

6. Comprehensive integration framework for all resources

Process activities are abstractions of real IT systems representing a wide diversity of application platforms, operating systems, object models, programming languages, and network locations. Some are packaged applications, some are custom; some web-based, some not; some batch, some interactive; some local, some remote. BPM links the process model with a consistent middleware framework that can interact with the data of any IT resource, despite this diversity. It provides a consistent process interface to the full panoply of EAI connectors, message queuing and broker middleware, and data transformation tools. BPM takes EAI technology and applies it to long-running end-to-end processes, including human interaction and B2B communications.

7. Modeled exception handling and transaction recovery

Process models must be able to handle a wide range of real-world *exceptions*, including both *system faults* – illegal data values, server crashes, and communications failures – and *business exceptions* – an account is overdrawn, a customer cancels an order in-process, or a trading partner fails to respond within a prescribed time limit. Process logic should include explicitly modeled *exception handlers* that can listen not only for system exceptions, but also for external event messages (e.g. user cancellation) or timeouts while a process is running, and invoke an exception process whenever an exception or event occurs.

In some BPM offerings, exception handling and transaction recovery logic is embedded in the resources themselves, rather than performed by the process model. In a sense this violates the core principle of externalized process logic. Exceptions can occur at any time in the process, and the required handling action is in general state-dependent. At a minimum the exception must not “break” the process but either correct the exception, retry the activity, terminate the process “gracefully,” while providing consistent state management across all nested and concurrent subprocesses.

In many cases, the process model must be able to group process activities into transaction units that succeed or fail “atomically,” meaning either all or none are executed, using some transaction recovery mechanism. Because business transactions may be long-running and not all resources are transactional, classic transaction rollback using two-phase commit or similar protocols is frequently impossible, so process models should also support recovery based on *compensating actions* and alternative paths of execution. The exception process should be modeled using the same tools as the normal process, bringing exception handling, compensation, and transaction recovery within the umbrella of the end-to-end process model.

8. Trading partner integration

In addition to automating and integrating end-to-end processes within an enterprise, BPM must also link the process with trading partners, who typically represent both the *customers* of the process and the *providers* of certain process activities. This involves features such as public process interfaces, trading partner profile management, and support for important B2B communications channels such as EDI, RosettaNet, ebXML, and conversational web services.

9. Business activity monitoring and reporting

At runtime, BPM must be able to track not only the state of individual process instances but process activity as a whole – e.g., the current value of all orders in process, the average order-to-ship time, or the percent of all orders fulfilled without human intervention. *Business activity monitoring (BAM)* implies the ability to translate IT events into business events, aggregate data from diverse sources, combine that data in *key performance indicators*, and present those metrics graphically via a real-time management “dashboard.” Ideally, BAM goes beyond historical reporting, and senses exceptions and deviations from key performance indicator targets in real time, generating alerts and remediation actions automatically.

BPM 2.0 Requirements: Process without Programming

Implicit in the basic BPM vision has always been the goal of enabling business analysts and integration specialists who are not skilled programmers to create executable models implementing real-world business processes. Today, an increasing number of BPM products are beginning to realize that goal in a new generation of software we shall call BPM 2.0. BPM 2.0 layers on top of and extends the core functional requirements described in the previous section.

While the core BPM requirement is for graphical design tools, many products still assume that modeling is performed by programmers. Key process activities require snippets of Java code, for example, or custom business objects must be created by developers. In BPM 2.0, the modeling tool is designed for non-programmers. Common functions like specifying flow routing or basic actions do not require code. Where code is required, such as in custom web user interfaces or complex system integration, developers work in the *same modeling environment* used by non-programmer business analysts and integration specialists.

1. Unified modeling and management environment for end-to-end processes

In some BPM offerings, the overall process model combines a workflow model of the long-running process built with one tool and an integration model representing application-to-application data flows, built with a completely different design tool and modeling paradigm. There may even be additional modeling environments required for message broker flows or B2B flows. Often, these tools represent different modeling languages, data models, and

business objects, forcing process designers to translate and transform process data between them. Also, they often make it difficult to aggregate runtime events from all the different types of process engines in order to create a true end-to-end BAM environment.

BPM 2.0 goes beyond the core requirement to demand a unified modeling environment for the entire end-to-end process, including EAI flows and human workflow tasks. Regardless of how the automation is implemented at runtime, the modeling environment provides a consistent abstraction layer for all types of process activities.

2. Service-oriented architecture

BPM 2.0 simplifies integration by modeling the functions provided by external resources and processes as *services* invoked by the process model. Such a service-oriented architecture hides the underlying implementation details of these diverse systems behind a consistent framework of service requests, replies, and events provided by the BPM software infrastructure. In BPM 2.0, activities in a process model just make service requests, listen for service replies or events, and control the flow of control in the process model. This simplification is what makes the integration of diverse applications and systems in end-to-end processes possible. It is important to note that *service-oriented* does not mean *web services-based*. In fact, most BPM 2.0 offerings provide their own service abstraction layer on top of proprietary connectors and integrate multiple communications protocols, including Java, CORBA, XML messaging, and web services. A pure web services approach – call it BPM 3.0 – may ultimately provide a single standard protocol that works across all resources, and eliminate the need for proprietary connectors.

3. Point-click service request modeling

Combining the above three requirements, BPM 2.0 offerings are evolving toward a consistent picture of what process modeling “looks like” in the design tool. Supported by EAI connectors, the designer can examine or *introspect* the list of service operations and events supported by any resource, such as an ERP system or DBMS, and select those needed for the process model. Each operation or event specifies a particular *request* (and in some cases, *response*) data structure or *schema*. Using the graphical data transformation tool provided by the BPM software, *transformation mappings* are defined between *process variables* (data carried by the process instance) and those schemas.

Having populated the palette of required connector-enabled service requests and data transformations, the process designer can build the process model. The process diagram interconnects nodes representing *elemental actions*, such as service requests or waiting for events, and *flow control constructs*, such as conditional branching, looping, or parallel splits and joins. Each action node is configured by point-click dialogs, binding it to a particular service request, schema, and connector from the palette, and by assigning a *mapped process variable* (i.e. a selected transformation on a selected process variable) to supply data values to it. All of this is point-click selection in the process design tool. This will be illustrated in detail in the product drilldown chapters.

4. Solution value out of the box

Real-world business processes are inherently more complex than the basic sample or tutorial applications provided with modeling tools. At the same time, the basic flow logic, communications protocols and exception handling procedures, business objects and vocabularies, and data transformations for a particular vertical process such as broker trade settlement, HIPAA claims, or RosettaNet order management may be largely the same across user implementations. Since, all other things being equal, users would generally rather buy solutions than tools, providing vertical process solutions, including pre-built flows, objects,

transformations, workflow task user interfaces and special runtime components that accelerate deployment of real-world implementations are extremely valuable.

However, buyers today are insisting that such packaged solutions be fully QC'ed, documented, and supported by the BPM vendor, with a commitment that as the underlying BPM platform and tools are enhanced, the solution will be upgraded as well to support the new versions.

Reality Check

How well do real products implement the core BPM functional requirements? How much of BPM 2.0 is available today? In both cases, quite a bit, but it varies from vendor to vendor, and the devil is always in the details. The overwhelming breadth and complexity leaves IT managers interested in BPM technology unsure of how to begin to evaluate it.

Bruce Silver Associates looked at five leading vendors representing a variety of approaches to BPM, from integration-centric to app server-centric to workflow-centric to web service choreography to hybrids combining all of these. Our review involved analyzing over a thousand pages of documentation on each product, and distilling it into an in-depth review of capabilities and process modeling and management procedures in a consistent framework applied across all BPM products. More information on the full report is available at www.brsilver.com.

Vitria BusinessWare 4.1 is reviewed in depth in a 60-page chapter of the report. The following is a summary of the report's conclusions, highlighting BusinessWare's strengths and mapping the software against the checklist of BPM functional requirements.

Vitria BusinessWare

Strengths

Unified Modeling and Execution Environment

The BusinessWare Modeling Environment (BME) provides a unified graphical tool for "solution-level modeling" of end-to-end processes, including application integration, data transformation, human workflow, B2B, web services, and activity monitoring, and a single BusinessWare Runtime Environment for executing them. This greatly simplifies modeling and state management compared to offerings in which separate models for application integration and workflow, for example, must be stitched together. Moreover, the modeling environment supports an integrated debugger that allows the designer to set breakpoints in the process diagram, inject test events, and examine data values. It also provides wizards that automate deployment of the process to runtime servers.

Component Integration Architecture

BusinessWare 4 supports flexible integration and reuse of modeling objects by decomposing end-to-end processes into major service-oriented building blocks called *components*. Integration models show how all components fit together in the end-to-end system, providing a high-level view of the most complex processes. Any process, subprocess, integration model, or connector can be converted into a component in the BME and exposed for integration *as a service*, either via J2EE or as a web service. BusinessWare generates the runtime proxy for the component service automatically. Libraries of components can be stored in a BusinessWare repository for easy reuse in multiple processes.

State Management and Exception Handling

All parts of a BusinessWare process, from top-level components to low-level process models and data transformations, are part of a consistent nested hierarchy, in which the process state at every level is carefully coordinated. Exceptions – including system faults, timeouts, and business exceptions such as cancellation or change of an order – are communicated between parent and child processes consistently and automatically, regardless of whether the subprocess was invoked synchronously or asynchronously. All parts of the model maintain complete visibility of the process state, avoiding the workarounds for exception handling and state management required in many BPM products.

Business Analysis and Monitoring

While many BPM products provide graphical reports based on runtime process events, BusinessWare's Real Time Analysis capability adds a framework for *automatically triggered actions*. Process Query models continually monitor running processes in real time with SQL-like queries, such as the fraction of instances falling outside of service level agreement targets, and issue events when the query result changes. These events can trigger alerts to users, escalation actions, remediation actions, or process changes automatically, fulfilling the vision of active business activity monitoring in a way most other products do not.

Referenceable Customer Solutions

Vitria has been providing the full suite of BPM functionality, as defined in this report, longer than any other vendor. Thus a key strength of the company is its installed base of referenceable large-scale production installations, particularly in financial services, healthcare, and telecommunications. While this report has focused largely on modeling, runtime scalability and reliability are equally critical to implementation success, and referenceable customers provide the best evidence of capabilities there.

Mapping to BPM Functional Requirements

High-level functional requirements for BPM and BPM 2.0 were described at the beginning of this report. Here is how Vitria BusinessWare 4 stacks up:

Model-Driven Execution

Yes, definitely. Not only are all runtime components generated automatically by the process model, but BusinessWare provides model-driven testing, system configuration, and deployment, part of a capability Vitria calls *Solution Lifecycle Modeling*.

Externalized Process Logic

This is a key strength of BusinessWare. All process logic, including exception handling, is exposed in the process model.

Graphical Modeling Tools

Design of integration models, process models, and transformation models is graphical, and wizards simplify data modeling, process actions, and connector configuration as well. In many cases, developers can bypass the wizards and write Java code directly, if required.

Support for All Process and Activity Types

BusinessWare supports long-running processes, EAI flows, human workflow, and activity monitoring queries within a single model.

Nested and Chained Subprocesses with End-to-End State Management

As described above, this is an exceptional strength of BusinessWare.

Comprehensive Integration Framework for All Resources

BusinessWare integrates with packaged applications, legacy systems, J2EE components, and web services through a comprehensive suite of connectors and native integration objects (proxies). Integration is transport-independent, allowing the customer to leverage J2EE services, web services, or its existing investments in message bus infrastructure, as appropriate.

Modeled Exception Handling and Transaction Recovery

As described above, this is a key strength of BusinessWare.

Trading Partner Integration

Vitria provides a strong B2B component through its eServices Framework, including prebuilt components that support message transport, message packing and unpacking, sequence validation, grammar and schema validation, business handshaking management, and security services between trading partners using EDI, EDI-INT, or RosettaNet. Vitria also provides a trading partner registry and integration of B2B public processes, or Partner Interaction Models, with the end-to-end private process.

Business Activity Monitoring and Reporting

As described above, this is a major strength of BusinessWare. The software provides both real time and historical reports, and a graphical management dashboard that can provide both high-level and drilldown views of key performance indicators.

Unified Modeling and Management Environment for End-to-End Processes

BusinessWare achieves this BPM 2.0 requirement, and extends it with testing, configuration, and deployment driven from the unified modeling environment.

Service-Oriented Architecture

BusinessWare achieves this by encapsulating processes, subprocesses, and connectors as reusable service-oriented components that can be exposed for integration either as Java objects, EJBs, or web services. Individual activities in process models (process actions), however, are not modeled as services, as they are in BPM offerings based on web service choreography.

Point-Click Service Request Modeling

Integration of service-oriented components is largely point-click, as is the construction of types from existing business objects and web services. BusinessWare wizards, such as the Condition Builder and Action Builder, provide point-click definition of process actions, which as mentioned above are not service-oriented. However, while BME does not require much coding, Vitria does not try to shield the designer from the syntax of Java or IDL, and the BME is not really oriented to non-technical business analysts.

Solution Value Out of the Box

This is also a strength. Prebuilt industry solutions for healthcare and insurance, financial markets, telecommunications, and other vertical industries include application-specific process models, business rules, data objects, application transformations, B2B transformations and reports. Also, Vitria's CIM offering provides common business objects

for application integration within the enterprise, and prebuilt data transformation “blades” for major packaged enterprise applications.

Bruce Silver, September 2003

About The Manager’s Guide to BPM Software

Interested in a detailed walkthrough of Vitria BusinessWare? Check out *The Manager’s Guide to BPM Software*. The full report provides a deep drilldown into the technologies of process automation (workflow), application integration, and B2B integration, and detailed walkthroughs of Vitria BusinessWare and other leading product offerings, representing integration-centric, workflow-centric, app server-centric, web services choreography, and composite BPM architectures.

Written for IT managers, the report provides what vendor websites and white papers don’t – a consistent, comprehensive framework for analyzing and comparing BPM software offerings. It also reveals what other industry analyst reviews don’t – a sense of what each product actually looks and feels like to use, what’s out of the box, what’s point and click, what’s custom Java code. Based on actual product documentation, the *Guide* walks you step by step through modeling an “end-to-end” process, including the graphical flow definition, introspecting and configuring application connectors, specifying and transforming process data, modeling human workflow tasks, and setting up business activity monitoring.

The Guide contains over 450 pages, with over 250 screenshots and diagrams. And it’s up to date. The walkthroughs represent the latest version of each offering as of the publication date of September 2003. For more information and free downloadable content, go to www.brsilver.com.

About Bruce Silver Associates

Bruce Silver Associates provides advisory services for both buyers and vendors of business process management and enterprise content management technologies. For BPM buyers, we can help with needs analysis, RFP development, and vendor selection. Dr Bruce Silver is a well-known BPM/ECM consultant and independent analyst, the author of *The Manager’s Guide to BPM Software*, and a regular columnist for Transform Magazine. Prior to founding Bruce Silver Associates in 1994, he was Vice President of the industry analyst firm BIS Strategic Decisions (later Giga Information Group), where he directed services in the area of workflow and document management.

Find out more about our services at www.brsilver.com.