Business Process Management: The Third Wave

Howard Smith and Peter Fingar

ISBN 0929652339
Category: Business
Hardcover 311 Pages
Web site www.bpm3.com
Meghan-Kiffer Press

Technical Excerpt
BPML

Endorsed by the Business Process Management Initiative and the Workflow Management Coalition

About this Book

While the vision of process management is not new, existing theories and systems have not been able to cope with the reality of business processes—until now. By placing business processes on center stage, corporations can gain the capabilities they need to innovate, reenergize performance and deliver the value today's markets demand. This book heralds a breakthrough in process thinking that obliterates the business-IT divide, utterly transforms today's information systems and reduces the lag between management intent and execution.

A process-managed enterprise makes agile course corrections, embeds Six Sigma quality and reduces cumulative costs across the value chain. It pursues strategic initiatives with confidence, including mergers, consolidation, alliances, acquisitions, outsourcing and global expansion. Process management is the only way to achieve these objectives with transparency, management control and accountability. During the reengineering wave of the 1990s, management prophets' books full of stories about other companies were all you had to guide the transformation of your business. Although their underlying theories were based on age-old common sense and general systems theory proposed fifty years earlier, they offered no path to execution. By contrast, the process-managed enterprise grasps control of internal processes and communicates with a universal process language that enables partners to execute on shared vision—to understand each other's operations in detail, jointly design processes and manage the entire lifecycle of their business improvement initiatives.

Process management is not another form of automation, nor a new killer-app nor a fashionable new management theory. Process management discovers what you do, and then manages the lifecycle of improvement and optimization, in a way that translates directly to operation. Whether you wish to adopt industry best practices for efficiency or pursue competitive differentiation, you will need process management. Based on a solid mathematical foundation, the BPM breakthrough is for business people. Designed top down in accordance with a company's strategy, business processes can now be unhindered by the constraints of existing IT systems. Short on stories and long on insight and practical information, this book will help you write your own story of success. It provides the *first* authoritative analysis of how BPM changes everything in business—and what it portends. Welcome to the company of the future, the fully digitized corporation—the process-managed enterprise. Welcome to the next fifty years of business and IT.

Appendix A

The Language of Process

One of the images I have in mind when I contemplate the universe, is that it is constructed upon a simple pattern of order that may be seen in any and all phenomena, no matter how complex.
—Jonas Salk

Consider carpentry as a field of human activity. "Hammering," "sawing," "screwing," and "measuring," using "hammers," "saws," "nails," "screws," "screwdrivers," "glue guns," "levels," "measuring tapes," and "carpenter's pencils": these words form a vocabulary describing the operations that can be performed in this field, and the means for carrying them out. Now consider business processes as a field of human activity. Processes, process data, activities, messages, rules, computation, process branching, compensating activities, exceptions, sequences, joins, splits, operations, assignments, transformations, schedules, rules and time constraints: These likewise form part of a vocabulary describing the operations that can be performed in the field. The tools for realizing these operations are process modeling languages. These languages provide semantics for business processes and unify the different vocabularies of process development, system integration, workflow, human interaction and transaction management, much as blueprints help the architect and the carpenter find a common language that enables them to work together.

An Open Process-Modeling Language Standard is the Enabler

Today's global telephone system simply couldn't exist without standards. The Internet, the network of networks, couldn't exist without standard protocols. It's easy to imagine that if each computer company offered its exclusive version of network protocols or Hypertext Markup Language, the Internet would utterly lack its "inter," its role as a network that connects machines and processes all over the computing and geographical world, each of which may use different vocabularies to describe what they do. Plenty of proprietary network protocols and private networks were to be found on the market in the past, but the availability of an open, universal standard changed everything. The Internet is ubiquitous, radically inexpensive to access and use, but still allows companies to build all manner of commercial offerings on top of it. Netscape was the first stratospheric commercial offering built atop the new network of networks; many other such commercial projects have come along, and many have passed on, since. Their success or failure was determined by the marketplace, not by the technology of the Internet itself.

It's easy to imagine that if each computer company offered its exclusive version of a modeling language for business processes, chaos would reign and the market for process solutions would remain small.

An open, universal language for describing business processes would, and, we hope, will, provide a ubiquitous business process management platform atop which computer companies can build new value.

A complex mix of marketplace dynamics, set against the compelling mathematical foundation inherent in process calculus, will govern the success of any modeling language vying for attention in the marketplace. Some computer companies will naturally seek to enforce their will on the process battleground by carving out their own proprietary language of process, believing they can use their market power to secure competitive advantage and lock in their customers. In the long run, however, the marketplace always wins. The Internet itself bears witness to this truth, since no one company or group controls the standards that make the Internet the Internet. Those standards are open to all. But more importantly, the correct routing of Internet packets depends upon algorithms that no one can argue with. We believe that for the emerging business process automation industry to flourish and fulfill its potential, similar principles should apply to the definition of process standards. You can't buck math!

Although several process modeling languages have been developed or are on the drawing board, along with a raft of associated Web services standards proposals, we focus in this book on a business process language that was conceived and developed from the ground up to enable the third wave of business process management, and based on a nonproprietary, royalty-free specification open to all.

Other process-oriented language specifications have been devised based on individual paradigms such as the needs of manufacturing, task management, application integration and Web services orchestration. In addition, there is a considerable body of work within academia referred to as the ontology of process representation. It aims to meet the needs for solving problems in particular application domains, for example planning and scheduling. These roots are significant, for as we explained in Chapter 5, "... the business process itself needs its own technology, its own organizing principles and its own paradigm. Hacking at the holistic BPM paradigm by throwing a grab-bag of individual paradigms at it is just that—hacking." Likewise, pushing task management, integration or nascent Web services technologies beyond their natural limits by adding more features and layers to the products already in use in business won't provide the BPM *management* capability envisaged of the third wave—an

environment built from the ground up with a rich process entity at its core. A unifying and simplifying step is required, on which the members of the industry of "BPM technology suppliers" can build new products and new services.

If such systems are to succeed in the marketplace, they will be built on a strong mathematical foundation, as were relational database management systems (DBMS) before. To date, the RDBMS is perhaps the most successful of all enterprise computing infrastructure products. The mission-critical BPMS that end-user companies now demand will likewise stand on the shoulders of giants in the field of process calculus, workflow management, the ontology of process representation, concurrent programming, finite state automata, and other technical disciplines. Its success will, however, depend not so much upon whether it implements one standard or another, for standards-acronyms are only fully understood by members of esoteric committees, but upon the capability it provides to business—the BPMS' features, performance and robustness. This in turn depends not on the detail of individual standards, but upon adherence to the mathematics required for process management to work in practice.

Therefore, because we want to convey the comprehensive requirements for a universal language of process, and not because we choose to ignore other work, we focus our discussion on the Business Process Modeling Language (BPML)[1] published by the non-profit Business Process Management Initiative (BPMI.org). Based on process calculus, BPML pays close attention to the needs of companies wishing to build process-managed enterprises. We do not claim that other approaches are defunct or invalid. Nor do we claim with certainty that other new languages oriented to process management will not emerge in due course. It is possible that a number of process modeling languages will co-exist and that there will be a need for interoperability among them.

Fortunately the process paradigm provides an answer to interoperability: A process expressed in one language can become a participant in another process expressed in a different language. This is called a process interface. It, too, is a process. Therefore, BPMI.org can be expected to continue to develop BPML as a superset of the semantics required for modeling both end-to-end processes and process interfaces. The industry will benefit from interoperability and choice. Although some software vendors may have to provide support for multiple

languages for some time to come, those end users and vendors that understand the significance of the process calculus foundation and insist on it will withstand any changes in process standards that one or more vendors may impose on their customers.

The Business Process Modeling Language

BPML is a specification both for building process management systems and for modeling business processes. BPML provides the required abstract model for all processes, along with a standards-based XML schema and syntax for expressing and managing business processes. BPML is the language for process management, just as XML is the language for business data and HTML is the language for hyperlinked Web pages.

Although it is possible for business analysts and technicians to model and execute processes using BPML directly, process management tools built on top of BPML hide the details from non-technical users. The scenario is similar to the way HTML is used: Technically oriented people use it directly while business people use high-level tools built on top of it. Although most users of process management systems will never know that its foundations are steeped in process calculus, this mathematical foundation provides consistency and ensures that everything "works." The same assurance was achieved in the management of data through strict adherence to the relational data model and within data query languages built from relational algebra.

A key characteristic of BPML is that it is directly executable on an IT infrastructure. It relies therefore on the existence of an execution environment. This is not the same as rapid application development, where executable code is generated from a model—BPML *is* the executable code. BPML is executed by a "process virtual machine" within the process management system. This is comparable to the way a Java program is executed by a "Java virtual machine" provided by a computer operating system.

BPML defines only what is required for expressing business processes, not any details relating to the systems developed to run it. Therefore, software companies are free to innovate in terms of the performance, scalability, robustness or other aspects of their products that businesses may look for when comparing different process products

and solutions.

BPML defines just what is required to establish a standard for processes, just as the relational model defined just what was required to establish a standard for data. This means that BPML covers aspects such as business activities of varying complexity, business transactions and their compensation, process data management, concurrency, exception handling and operational semantics. BPMI.org does not aim to standardize the approach vendors take to the internal construction of a BPMS. Rather, it provides standards for the formal process model a BPMS should support and should expose to the business via process query languages and process design tools.

BPML processes are clear. The meaning of a BPML expression and what it will do when it executes are never ambiguous—a declarative specification. A BPML process transferred from one vendor's implementation to another both means, and does, precisely the same thing, just as transferring data between databases has no effect on the meaning of the data to the business that originated it.

BPML provides a vocabulary for enabling the persistence and interchange of process definitions across heterogeneous systems and modeling tools. This is crucial to businesses and the process industry alike. Companies want to build their process management infrastructure using best-of-breed components. Much as companies now procure a database from one vendor and applications from another, they will want to do the same for processes. They will also want to incorporate process models they have developed in the past.

BPML defines an abstract model and grammar for expressing any business process, Web service choreography or multiparty business collaboration. It does not define any domain-specific semantics, such as the details of supply chain logistics, enterprise resource planning or customer relationship management. Nor does it contain anything specific to any particular vertical industry.

BPML is the language of choice for formalizing the expression, and execution, of collaborative business interfaces. This includes both the data exchange requirements and the process interface by which data is exchanged. BPML defines process data using XML schemata, allowing industry data standards to be used in conjunction with process definitions. BPML can express the inclusion of the data definitions from an industry standard. An example would be, the content of a purchase

order exchanged by two process activities, together with the industry's standard procedure for doing this.

The industry has developed visual high-level BPML modeling tools that can be used to describe both collaborative and transactional processes, in a way that business people will understand. BPML is open to the whole community of process engineers, business analysts and system architects, helping companies coordinate and streamline the development of processes inside and outside the enterprise.

Users of BPML can share process descriptions without divulging the technical implementation details of their companies. The approach breaks the proprietary programming cycle that spawned internal e-business solutions that outside developers could not manipulate and business users could not read or understand. BPML provides the business confidence needed in collaborative commerce.

Business people need to be able to communicate in a way that is comfortable for them. The full potential of any standard language remains unrealized until business people can communicate using it. Like complex XML and HTML syntax, BPML is not designed to be easily readable by most business people. Therefore, BPML also has an equivalent graphical visual notation, the Business Process Modeling Notation (BPMN). BPMN uses a simple set of drawing symbols that represent BPML elements. Users manipulate the symbols—various geometric shapes, arrows, and the like—and link process flows graphically. Underneath its "graphic skin," the model is expressed in the form of BPML that more technical staff can use. To those old-enough to remember, BPMN resembles high-school flow-charting.

BPML and BPMN are unique in their ease of use yet powerful enough to develop sophisticated end-to-end processes. These processes can be as complex as traditional software-based applications. Much as spreadsheet programs provide powerful macro capabilities that allow advanced users to develop complex numerical models, though not all spreadsheet users use, understand, or care about these features, BPML can be used by all users, regardless of their level of "technical" expertise.

As shown in Figure A.1, there is a one-to-one correspondence between BPML and BPMN. The diagram represents the code and the code represents the diagram. There is no loss of information when moving between the two. The clear advantage is the ability to depict processes in a way that business users can both understand and execute.

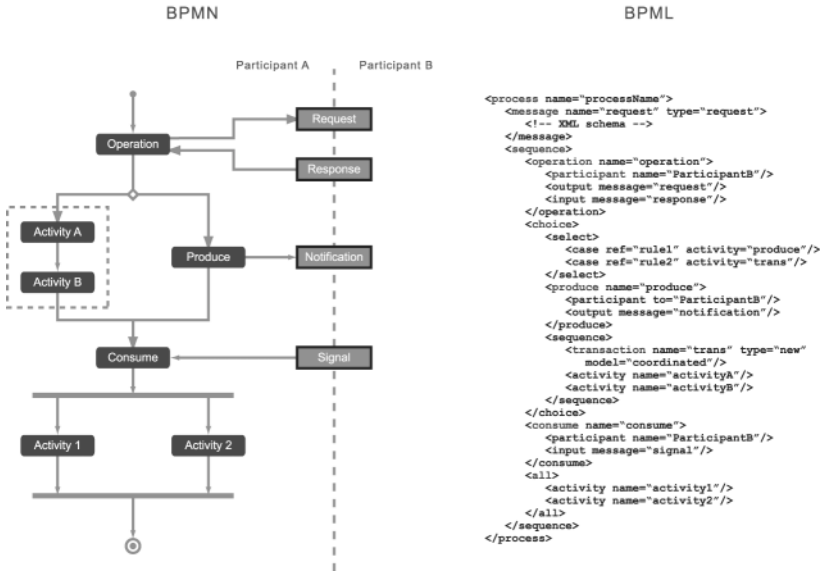

Figure A.1. The Business Process Modeling Notation (BPMN) and Business Process Modeling Language (BPML version .4, March 2001)

Although BPMN meets the urgent need for process communication on one level, many new and innovative notations for BPML are likely to be developed, built on the foundation that BPMN has established. For example, a project plan is a type of business process, and whereas plans *can* be represented in BPML using timing, resource and constraint attributes, viewing such a plan model in BPMN may not be the best choice; Gantt charts were invented for that purpose. Businesses can expect more and more commonly used business tools such as project planning software to be based—underneath—on process modeling languages. This will allow the planning process, in fact all business information, to be put directly into operation. If the plans a company develops using its preferred desktop tools are based on BPML, they can be directly executed across the entire business, and the company's business partners as well. The "live" plans will link to live systems, inform employees of work to be done, close the loop with managers and supervisors and provide everyone with a live view of project status.

The formal foundation of BPML provides the reliability, coherence

and simplicity needed for users to be able to manipulate processes with great confidence. In the past, deploying a new process (application) was fraught with danger and therefore subject to all manner of checks, balances, tests and validation. The risks of rolling out a new application that proved to be unreliable were so huge that weeks or months could pass before it was deemed ready for production. Business processes, on the other hand, are always valid and ready for use. Deploying them is no more complex than saving the design file. Removing a bad process is just as simple. This does not, however, mean that every process design will have the desired impact on the business. Simulation in advance of use is required for complex, high-impact processes. Much as aerospace firms subject their jumbo jets to intensive testing, businesses will put their high-impact processes through a workout before final deployment.

Because the focus of process management is often collaboration between departments or between companies, BPML offers complete support for distributed, concurrent, executable processes. BPML supports multiple process participants, allows for the production and consumption of messages, and supports dynamic process branching, transparent persistence, embedded business rules, nested processes, distributed transactions and exception handling.

In BPML, collaborative process models can contain as many, or as few, execution details as the process designer considers necessary to share between partners or business units. BPML also supports a high level of abstraction, in which execution details can be hidden. This approach promotes collaboration among business partners. A model of the process can be shared even as certain details of implementation can be left to each partner. This is a vital breakthrough for true interoperability, taking integration one step further, to true process collaboration.

BPML copes with the complexity of collaboration among business partners by permitting arbitrarily large numbers of participants to play a role in end-to-end business process designs, at any level of nesting and concurrency. Participants communicate freely with one other, right across the value chain. BPML achieves this by separating and interleaving control flow, data flow and event flow, while adding complementary and orthogonal design capabilities for business rules, security roles and transactions. Messages exchanged between participants contain the process data that is needed for process collaboration.

Tools that create BPML processes from users' actions create new

possibilities for innovative IT systems. For example, the concept of "assigning a task" can be defined in BPML. The business manager sends team members a message containing a description of a task, also expressed in BPML. Team members respond with a message indicating whether they can perform the task. The final message goes from the manager to the team member chosen to perform the task. Managers may then make their decision based on process data exchanged between the participants in these interactions. For example, the team members' acceptance messages could include their estimated time for completing the task. The manager could pick the team member with the lowest time estimate, or could examine records of past performance and select someone else. The entire logic of this process can be defined in BPML and shared across the business.

Similarly, in a telecommunications call center, customer service representatives do not think in terms of processes, but this does not prevent process management from forming the basis of minute-to-minute activities. When a customer service representative gets a request from a customer to configure a new circuit, he or she will not say, "What process do I need for that?" Instead, the service representative will interact with the network management system and instigate a process that will do everything necessary for the customer—with no thought given to the term "process," a new process nevertheless comes into being. This may itself be a complex and unique variant of a generic "customer pattern" designed especially for the individual customer, including everything required to ensure that the configuration of the network service meets that customer's requirements.

In a world of business process management systems, more and more operations on IT systems will be instructions to start, stop or interact within processes. Because the process management system has complete knowledge of the state and structure of all its processes, queries and updates can be constructed to modify processes in flight. Just as databases facilitate data transactions, process management systems will enable powerful manipulation of, and updates to, long-lived persistent processes. Extending this further, actions by users of BPM systems could generate dynamic processes "on the fly" that get put directly into operation. It's not an overstatement to claim that such capabilities are rendered technically trivial with the advent of BPML. The creativity and inventiveness of systems developers to read, write, generate, process and

manipulate BPML should never be underestimated. BPML is built upon the same set of powerful XML-based technologies that are already in use for a wide range of other purposes in every company and every software development shop on the planet. Those familiar with handling XML will be able to handle BPML.

The Impetus and Design Goals for BPM

In the middle to late 1990s there was an explosion of experimentation with new business models and business-to-business (B2B) integration over the Internet. Various standards proposals were put forward to address the problem of messaging and document exchange between business partners. The CommerceNet eCo Framework Project was the first and most profound examination of the use of XML for e-business. None of these initiatives, however, provided a way to manage a complete e-business scenario. What global corporation would open its back-office systems without a complete overview and a clear description of the processes governing their use? Everyone knew that something was missing. Some even named the missing elements, using words such as "orchestration," but it was not until after much practical experimentation had taken place that it became clear that process management was part of the answer.

Lack of a common language for expressing complex end-to-end business processes—not only the interoperation of services—was hindering the development and adoption of e-business. BPML was therefore developed by the BPMI in the spring of 2000 to provide a foundation for the new class of mission-critical enterprise infrastructure, the then abstract concept of the "BPMS." Such an infrastructure would no longer differentiate between internal and external integration: It would manage complete business processes, not just the public interfaces shared by business partners. In place of having to use separate tools inside and outside the enterprise—enterprise application integration (EAI) internally and B2Bi externally—companies could use one formalism to express all processes in the value chain, paving the way for the development of systems that could execute such processes directly.

BPML was indeed a radical innovation: a language that could model business processes in a way that would translate real-world ideas—the objectives of CEOs, supply chain managers, and strategists—directly to

operations, empowering the business user of the system. The language was based on a concept called "design-driven architecture" (DDA). In such an architecture, the "executable" elements are rendered, not as software, but as "data" that is interpreted by an external "engine." The "data" schema (the design) is chosen to perfectly match the intended purpose of the "engine," in this case, the representation, execution and management of business processes. A spreadsheet uses this technique, with rows and columns representing formulae "executed" by the spreadsheet program. The user simply models what is required and the model runs.

Processes such as order management, customer care, demand planning, build to order, product development and strategic sourcing were among the types of end-to-end processes the BPML development group wanted to be able to represent. They recognized that to achieve this would require extending the design-driven principle even to distributed concurrent computation. Their vision was not only to enable "e-business," but also to establish a permanent new foundation for all business systems. This approach, they felt, would make IT infrastructure much more adaptable to the actual, real-world, messy business processes inherent in any enterprise. It could lead to tools that would facilitate the management of constant change. It would be a step-up for the IT industry itself, complementary to the progress that had been made in distributed object based systems.

The developers of BPML realized that to succeed, such a language would need a rigorous mathematical foundation for the new "process data" so that systems built upon it could be as resilient as any existing mission-critical system, such as databases and ERP. They chose to employ a declarative semantics based on process calculus and a concurrent-processing model, so that process management systems could provide end-to-end process analysis, prediction, simulation, metrics and visibility across all participants. Although the mathematical theory may be difficult to understand, as is the relational algebra of the database, the benefits are easy to appreciate.

The Business Process Management Initiative developed rigorous design goals and a charter for the proposed representation and system. These included:

Bridge and unify internal and external integration. In order to unify internal and external integration, the Business Process Management

Initiative sought to sharply distinguish the concept of an end-to-end process design from the concept of an interface process design. This functional separation allows each design process to be managed independently, and supports the fluid movement of process work among partners, systems and their security firewalls. The creators of BPML found that this distinction added to, rather than detracted from, the security of e-business, since the rules for permissible, coordinated interactions between partners—the logic governing the flow of business information and the enforcement of business collaboration agreements—would be encoded in the processes themselves, not in technical systems interfaces.

Consolidate human-oriented workflow and machine processes. Previous process-oriented approaches assumed a sharp distinction between the work done by humans and the work done by computer systems. The developers of BPML recognized that this distinction had to be eradicated if such technologies were to "cross the chasm" and become mainstream elements in IT architecture. The workflow community, experts in this area, eagerly joined in to ensure this was the case.

Exploit the growth of hosted services. Web services technologies render "software as a service," a concept that has generated significant interest in the possibility for remote delivery of applications and business services over computer networks, and given rise to the application or business services provider (ASP, BSP) . In recognition of the potential of networked service delivery, BPML was designed to allow the inclusion of hosted services within process definitions and to describe the behavior, discovery and advertising of such services. The developers expected loosely coupled, hosted and network-resident services to play an increasingly important role in how businesses build, source and interact with outsourced third-party business services.

Include key back office systems. Recognizing that legacy systems contain millions of lines of potentially reusable code, BPML was designed so that it could accurately represent existing software and procedures. In this way, past investment in software development could be protected and leveraged by allowing existing procedures to be reused in the design of new processes. The largest, most complex, legacy systems can be tapped for their functionality, and even the tiniest, fine-grained application components can be exposed to participate in business processes. A future refinement of this capability is to automatically allow the

representation of standardized packaged applications as business processes, a technique known as *introspection* and *process projection*. This technique is already available for some standard packages.

Build on, rather than replace, middleware. Over the years companies have acquired a variety of infrastructure systems, including messaging, transaction processing, integration brokers, database management systems, object request brokers, directories, rules management systems and a variety of system management tools. More recently they have acquired application servers, enterprise application integration hubs and rules engines. Recognizing the value of prior investments in middleware infrastructure already in place, the BPMI group designed BPML so that new process management systems could sit atop standard middleware components, such as application servers, directories and messaging. The idea was that such existing investments would be integrated once, and only once, to the process management system. Thereafter, numerous processes could be developed and deployed in a stable IT environment with no need to "integrate" individual processes "technically."

Support business change. Based on an analysis of existing process modeling approaches, tools and methods, the design of BPML permits it to represent processes that can react and adapt to changing business requirements in real time. The designers recognized that real-life "business processes" were very different from the "applications" with which companies were most familiar. A real-life business process might evolve, amoeba-like, in quite unexpected ways during its lifecycle, acquiring new participants and new capabilities. BPML makes such "live" structural evolutions independent of the initial process design. The evolving structure is open to analysis by other computer programs.

Distributed execution. Because BPML was designed to be mission-critical, it had to be highly distributed: spanning systems, departments, businesses and even industrial value chains. Distributed "agent-based" architecture was key to the design: Each participant in the end-to-end process is a fully autonomous process. The processes had to operate correctly even if different participants in the process were running on different computers and using different process management software. The design challenge was not merely to enable distributed execution, but also to allow a distributed process to be managed from a single management console. Two different types of distribution had to be supported: distributed execution of the process, on multiple process

management systems, and a distributed process management system (for example a cluster or federation), with flexible control of process deployment and partitioning.

Exchange and reuse processes offline. The founders of the BPMI believed they were building the foundation for a new "process industry," and they looked forward to the time when there would be many suppliers of both new processes and new process management systems. Naturally, the full implications of this could not be foreseen, but the essential requirements for the exchange of adaptable packaged processes, both offline as well as online (peer-to-peer), were taken into account. Conventions for this were borrowed from the emerging XML standards. This approach allows complex process data to be freely exchanged and manipulated by commercial and open-source developers alike. For example, a process marketplace might emerge in which processes are sold on CD-ROM or downloaded from a file-swapping Web or FTP site. The BPML developers felt that corporations needed a similar mechanism in-house for sharing "best practices" internally and with partners, and that this would be one of the first applications of a process sharing service.

Reuse processes. BPML was designed so that processes could leverage existing process patterns, enabling a high degree of reuse. Drawing on lessons learned from previous approaches to software reuse, BPML emphasizes composition and adaptation, rather than specialization and generalization. One key goal of process-level reuse is to lay a foundation for the "process factory" and the process manufacturing industry. They sought to enable the rapid assembly, adaptation and packaging of new processes from pre-existing process patterns, much as industrial engineers reuse elementary shapes and objects to build complex three-dimensional product models.

Enable the combination of best-of-breed solutions. No single software company can provide a complete solution or be best of breed in all areas. BPML was designed to allow straightforward integration of commercial off-the-shelf (COTS) products. BPML was also designed to help software companies develop complete best-of-breed BPMS solutions, combining visual modeling tools, process engines and simulators, a trend similar to the emergence of the so-called fourth-generation languages (4GLs) in the field of data management. In this sense, BPML is as adept at integrating processes and applications at run time as it is at integrating

software products at build time.

Build on, not replace, existing standards. The BPMI charter recognized the importance of using existing standards and technologies, including standards for the exchange of information and events, business transactions, service advertising and discovery, real-time collaboration and Web services. BPMI adopted the philosophy of "federating" useful standards, and its design methods adhered religiously to this principle. BPML binds to existing and emerging standards, eliminating the fear people naturally have that the standards they have invested in may become obsolete. There are limits to this strategy however. Since much is at stake in the "standards-making" process, the future of process standardization cannot be accurately predicted. Fortunately, this is not an overriding concern for many end user companies who are simply looking for better tools with which to manage their business. Nevertheless, instability in standards can hinder the uptake of new ideas. BPMI insisted on an open process, balanced with the need to get work done and achieve results.

Orchestrate Web services. During 2001 and the early part of 2002, Web services became the marketing banner under which the IT industry rallied a number of important new standards for service interoperability. Considerable efforts were made by BPMI to ensure that process standards and Web services standards were compatible. Web services technology is not a new concept however; it's the convergence of service-oriented architecture—a software trend that began with object orientation in the 1960s—and Internet technologies that began in the 1970s. Web service technology is the latest manifestation of a long line of distributed computing architectures such as Microsoft's DCOM/COM+ and the Object Management Group's Common Object Request Broker Architecture (CORBA and CCM).

Web services technology is important because it removes proprietary barriers to application integration and provides a low cost, universal method for building composite applications. It is to business-to-business interaction what the World Wide Web itself was to business-to-consumer interaction: a radical and low-cost simplification. However, it is not a panacea solution, by any means.

Users of BPM do not require applications to participate using Web services. An over-emphasis upon Web services standardization misses the point that developers of BPM products may wish to develop their solutions in different ways internally, yet still expose a standards-based

capability to the business buyer. Pushing Web services standardization too far limits the potential of the industry to innovate, creates overly complex "standards stacks" and leads to "standards battles," since so much is at stake for vendors who choose "the wrong standard." BPMI therefore envisaged the use of Web services as primarily assisting the development of a standard process interface at the business process level, rather than imposing a standard approach to the development of a BPMS.

Apart from the radical reduction in costs they offer, however, Web services suffer the same problems as the EAI solutions they replace. Both EAI and Web services are focused on applications, not business processes. They share an IT, not a business, heritage and bias. They share the Application Programming Interface (API), not the business process, as their conceptual "first-citizen" foundation. The commonly held bottom-up view of Web services is that existing, discrete, components can be combined to form larger composite services and that these can eventually attain the status of a complete business process. This is not BPM. BPM is a new method and new technology for manipulating processes as a whole, not just composing services to create composite applications.

The top-down "whole system" view of BPM says that although Web services are useful as components of a process, the process design and its management must be oriented by conscious, business-led decisions, themselves often processes. BPM includes metrics, computational logic and business rules that can be embedded in process definitions and can constrain processes in accordance with business goals, such as limiting time, cost and resources. Web services are not inherently suitable for handling these process-oriented tasks, and how this family of technologies will evolve is anybody's guess. Do not confuse a software development paradigm, Web services, with BPM.

In reality, however, software developers have choices. Web services are an essential tool for application integration because they provide an intermediate step in the transition from traditional business applications to third-wave business processes as an organizing principle for software. BPML and Web services go hand-in-hand, as do bottom-up and top-down process design. From the point of view of Web services, the system designer's aim is to achieve more flexibility about how services and assemblies of services can be used. From the point of view of proc-

ess management, the systems designer's aim is to let the process management system establish a common environment in which all services are visible and available for reuse in processes. These subtle distinctions are hard to understand initially, but over time and as process engineers and software engineers work together and gain more experience, they will gain a full understanding of the roles process engineering and software engineering will fulfill.

A Universal Process Language

Business processes are perceived and described in different ways by people with different roles within the business. Software engineers, for example, understand business processes at the level of software implementation. They use a variety of notations and methods such as the unified modeling language (UML) for representing processes as software objects. Business people, however, understand processes at the level of material flows, information flows and business commitments. Often they don't think about technology at all unless it has become a routine element in their business.

Business people imagine ready support for integration and automation and often do not understand why technicians cannot make these happen easily, right away! They don't understand why integration at all levels—people, systems, processes and businesses—is so hard to achieve. Enter the business analyst, someone who bridges the gap between the non-technical business person and the software technician. He or she uses a variety of business architecture and modeling methods and tools to represent business processes, such as the Zackman Framework, IDS-Scheer ARIS and Computer Sciences Corporation's Catalyst. Analysts understand processes in terms of organizational coherence and business outcomes. They use a variety of methods to manage and improve process, including Six Sigma, TQM Activity Based Costing and reengineering. They have a very different perspective to technicians. They deal with "the process" more directly, whereas the software engineer deals with it indirectly, through the distorting lens of software artifacts. BPMI sought to change this. Their answer was a single process model, BPML, shared by all disciplines.

Business analysts are also experts in the processes of specific industries and participate in industry standards initiatives, such as the

standards developed by the Telemanagement Forum for Service Provisioning in the telecommunications sector, or the architecture for Straight Through Processing (STP) developed for the financial sector. Sometimes these models are developed out of a need to facilitate collaboration in the industry. Examples include the SCOR reference model developed by the Supply Chain Council; the Collaborative Planning, Forecasting and Replenishment (CPFR) protocols of the VICS committee, made up of retailers, manufacturers, and solution providers; the high-tech industry's RosettaNet Partner Interface Processes (PIP); the manufacturing industry's STEP framework for product lifecycle management; and FpML (Financial Products Markup Language) for securities trading. These standards are essential to business process collaboration, and BPML is able to model and describe all of them. The implications are far reaching. Industry experts can now turn paper-based specifications into executable processes, no matter what role or skill set they have.

Government regulations and the guidelines set out by professional standards bodies also drive demand for formalized process frameworks. Companies are required to implement and manage such business processes without adding to the cost of their products and services. These industry frameworks create many different perspectives on the nature and management of business processes in those industries. Process management systems and languages can now play a major role in helping companies reduce the cost and complexity of implementing mandated processes. Executable process languages such as BPML help align legislation and best practices with process models.

Companies in every industry probably feel that their own processes are unique. Yet each shares a single underlying model of behavior, but expressed in the standards of its industry. For example, in the retail sector, processes span many business partners, and are oriented to the sharing of information up and down the supply chain. In the financial sector, processes are more centralized: Often one internal process is used to receive and send messages. In the telecommunications sector, processes govern the day-to-day operation and configuration of service and network elements, responding to requests from customers for new services and providing an end-to-end view of all of the network infrastructure elements and their operational support systems (OSS).

Rather than trying to create different solutions for each industry,

the BPMI group designed BPML to adapt to and amplify a variety of standards. CPFR, STP, SCOR, HIPAA, RosettaNet, STEP, FpML, GAMP and CIDX can now all be expressed using BPML and be adapted for use by businesses within those sectors. BPML unifies the semantics of a number of different approaches to process definition, making possible the implementation of a universal virtual machine that can execute, literally, any business process.

A Rich Language for Process Engineering

BPML is the meta-language of process, a language for developing process languages that capitalize on BPML patterns. BPML can be adapted for use in different industries by assimilating the required vocabularies. Much as lawyers use specialized legalese and refer *by name* to individual statutes of bodies of case law, BPML can accommodate specialized languages to refer to *patterns* of process design. This will be vital to industry standards groups establishing best practice processes, but using their own vocabulary to do so. Every element and every design pattern in BPML can be named and referred to explicitly using their native vocabularies. An industry can therefore develop its *vocabulary of processes*.

BPML is rich enough to represent and express material flows, information flows and business commitments:

- *Material processes* transform raw materials or components into subassemblies and finished products, using resources such as people, machines and computer systems. These are based on the traditions of industrial engineering, whose key concepts are assembly, transformation, transport, storage and inspection.
- *Information processes* describe the storage, retrieval, manipulation, display and communication of structured and unstructured data and knowledge. These are based on the traditions of computer science and software engineering, whose key concepts are sending, calculating, transacting, invoking, deciding, saving, forwarding and querying.
- *Business commitments* and relationships articulate and satisfy conditions in interactions between customers and partners. These are based on structures of human communication and cooperation found in all languages and cultures. Key concepts are requesting, promising,

offering, declining, proposing, canceling and measuring.

BPML excels at representing discrete, distributed, transactional, computational and collaborative processes in any field, whether material, information or commitment based. The language is used to express the manner in which participants (people, systems, data, applications, trading partners and marketplaces) work together to achieve common business goals. This means that employees and business partners can achieve a common understanding of terms in the domain, such as *activity*, *exception*, *message* and *choice* as they work together. In fact, any set of terms can be defined and associated with BPML design patterns.

BPML has rich abilities for expressing business logic, control flow and information flow. Participants in the process can be back-office systems (such as a database or ERP system), software components (such as a Java component or Web service), users (such as a purchasing manager) or partners (such as a supplier or customer). Business transactions (for example the fulfillment of a purchase order) and system transactions (for example a transaction on a database table) can all be defined as part of a process. *Business* transactions usually involve two or more partners (e-business), while *system* transactions can involve multiple back-office packages (distributed transaction).

Processes can be defined to bind participants closely together, as in straight-through processing, or to be loosely coupled, as in supply chain management. Loose coupling is especially important in collaborative commerce, because partners will almost never share the same applications and systems, and they will certainly not allow others to gain direct control over their mission-critical back-office systems. Such processes involve rules, roles and renewal procedures and place fewer restrictions on participants with respect to the order in which activities must be performed. Few existing technologies support this kind of loosely coupled, reliable, behavior, and fewer still support it across different technology platforms and network protocols. BPML does. Each of these characteristics is essential, however, if collaborative commerce is to become widespread. BPML frees companies from such constraints and allows them to deploy the innovative new business models that they need to compete—as tightly or as loosely coupled as necessary, but with the required level of coordination and control.

These collaboration features are part of the process design model

itself, not an attribute of a hard-wired application. For example, a product manufacturer can, using BPML, define the way in which customers obtain all required product support services: consulting, configuration, fulfillment, installation and training. Each of the partners providing these services can operate quite freely but in the context of coordinated processes defined by the product supplier. The product manufacturer's end-to-end process model would describe just the coordination required between partners to provide the integrated service to the customer. As experience with the process deepens, it can be tuned to improve the customer experience, and to reduce cycle times.

BPML enables an unlimited degree of nesting, parallelism and concurrency in process design, and can accommodate both process-level transactions and compensating transactions. Think of a process transaction as the equivalent of a database transaction, but right across some subset of an end-to-end process. It has business meaning, quite independent of any "technical" transaction requirements. Furthermore, processes can be designed to be both reliable and self-healing. If constraints or restrictions are required by a business relationship, limits can be placed on the data exchanged by participants, the time spans of activities and the availability of services and participants.

Recognizing that the value chain is always going to be heterogeneous, the developers of BPML designed it to encompass the process semantics inherent in any other process language. BPML can be used to build bridges and process interfaces to other systems using other process modeling formalisms. For example, a company can use BPML to define its order-to-pay process even if another system, based on another process language, was used to define the order requisition part and yet another was used to describe invoicing. BPML can serve as a *lingua franca* for process languages. It should be possible to express any process in BPML that can be expressed in any other process language. Test this assertion in practice.

BPML simplifies interactions between processes running on disparate systems and across business domains. It has to exist in the heterogeneous, distributed, computing environment because that's real life. BPML processes can model anything that occurs in a typical Fortune 500 network infrastructure, including the different processing paradigms inherent in complex IT systems such as teleprocessing monitors, remote procedure calls, object request brokers, and publish/subscribe

messaging systems and queues. The unification of the dual notions of "business" and "technical" process played a major role in its design.

A Foundation for Collaborative Commerce

A business process involves two or more business partners. Each partner brings many "process participants" to the table, usually comprising the partner's back-office systems, e-business applications, employee interactions and other third-party elements. The process systems deployed within, or extended toward, each partner are responsible for the management of that partner's process participants. Thus the process management system can be seen as a gateway—a process-level firewall—connecting business partners that each have their own process management system or systems.

For collaborative commerce, the expression of a logically distributed process alone would not be sufficient. Even a single process with a centralized business design may need to be executed across a set of physically distributed systems, including systems owned by different companies. Because business partners normally interact with a large number of suppliers, customers and trading partners—hundreds, or even thousands—companies often need to establish a common "interface process" in order to simplify the required *technical* integration at the network level. To this end, BPML supports the concept of "public interfaces" and "private implementations."

A process deployed by a company on its BPMS usually instantiates the private implementation of a larger e-business process involving business partners. Partners participate by interacting through a public interface, usually defined jointly by both sides. For example, the Partner Interface Processes (PIPs) defined by RosettaNet and the Uniform Code Council (UCCNet) , comprise a combination of standard data and standard interface process. This enables the reliable exchange of data in the context of commercially binding business commitments. BPML systems generate the required interfaces for collaboration to occur. In a collaborative purchase order management process, for example, the enterprise's private implementation can be described as a procurement process while the supplier's private implementation of the same e-business process can be described as a fulfillment process. The enterprise's procurement process and the supplier's fulfillment process are

two private implementations of the same end-to-end business process. These implementations interact with one another through a common public interface.

The private implementations and public interfaces of an e-business process give rise to corresponding notions of "private processes" and "public processes." But do public processes really exist? The BPMI advocates the concept of a private process, but only as a restricted case of a business process that does not involve a participant other than the enterprise itself—in other words, a strictly internal business process. Moreover, the notion of a public process makes sense only if a single entity is responsible for its execution—for example, an EDI network, an electronic marketplace or a process service provider (e-hub). Even in such cases, however, the hub or marketplace is not really the entity responsible for the execution of some imaginary public process, but rather a business participant in the overall end-to-end process, with real internal processes of its own.

The private implementations of an e-business process are the only parts of a business process that must be executed. Its public interface is nothing more than a vector for the collaborative execution of the e-business process by its participants. Such a public interface is not executed independently but rather consists of the various private implementations of the e-business process—one per participant. It's like looking at the behavior of internal processes from the outside, in terms of their interactions with others.

Business interfaces change infrequently—to align with industry standards for example—and yet companies must be free to continuously innovate along the entire length of the end-to-end process. Although participants in a collaborative process may be internal or external, a BPML process model does not distinguish between the two types. Process systems can be developed that will allow flexible and secure process outsourcing or the division of responsibility among partners, while maintaining a consistent business interface. In fact, the business interface can itself be described as a process and used as a pattern in the design of the end-to-end process, allowing each to evolve at its own pace.

In some cases the interface may itself be a complex process. For this reason, the BPMI has developed BPML-based languages to allow interfaces to be defined independently of end-to-end process designs,

using what is known as a *process interface definition language* (PIDL). The messages exchanged between any two process participants constitute a process and can be explicitly modeled as such. BPML itself can be used for such a purpose, although other languages for doing this have been and will continue to be developed; their requirements being not as complex as for an end-to-end modeling language.

One way of visualizing this to imagine a virtual participant sitting between two (or more) real participants; the virtual participant being responsible for mediating between the internal processes of the others. Or think of a Coke machine. On the one hand there exists the end-to-end process of obtaining a can of drink and consuming it. On the other hand there exists the coin slot and drinks dispenser—the interface between the machine, with its internal processes and the human, with his or her internal processes that enables the transaction known as *satisfying one's thirst*. BPML can describe the whole process, including the behavior of the interface. For example, BPML would define the fact that it is possible to put a coin into the slot, but not take one out, and that the converse is true for the drink dispenser. BPML could also define the fact that the human could obtain a drink if and only if he or she deposited the proper number and type of coins. On the other hand, BPML could also describe the private internal processes of the user and the machine: the raising of the arm to put money in the slot and the mechanism inside the machine that drops a can of soda into the bin. Similarly, the enterprise's procurement process and the supplier's fulfillment process are two private implementations of the same end-to-end business process—which we could call "order to pay"—that interact with one another through a common public interface. The interface could adhere to an industry standard such as a RosettaNet PIP. The BPML approach allows separation between different parts of the order-to-pay process —requisition and invoicing. The company can also describe the flow of messages required to process an incoming purchase order, the exchange between buyer and supplier, independently of larger processes in which such a reusable pattern is required.

Both illustrations—the Coke machine transaction and the order-to-pay—are business processes. Each can be modeled, described and executed in BPML. The distinctions between end-to-end processes, internal and external processes, interfaces, real participants and virtual participants are subtle. The point is that any fit-for-purpose process modeling

language should be able to model everything required for end-to-end management and execution. For e-business integrators the advantage is the ability to implement an interface once, covering all participants, without limiting in any way the ability of those participants to collaborate across the boundary. Even if an industry requires a particular interface process—for regulatory, security or other reasons—or has developed an effective one in the past, companies can go ahead and use it. If it is already working and in use, BPML can adapt to it.

To make process design practical for collaborative commerce, users will either establish a common process repository and management system, or will support the import of processes to local tools allowing modifications by partners. Sophisticated repositories are not always needed and could be regarded as overkill. Processes can be freely shared using a variety of mechanisms including File Transfer Protocol (FTP), the Web and e-mail. Distributed authoring and versioning of business processes is already supported in process management products using the widely adopted WebDav (World Wide Web Distributed Authoring and Versioning) protocol published by the Internet Engineering Task Force (IETF), a standard for collaborative authoring on the Web.

WebDAV is a set of extensions to the Hypertext Transfer Protocol (HTTP) that facilitates collaborative editing and file management between users at separate locations. WebDAV is expected to have an impact on the development of virtual enterprises by enabling remote groups to work together in new ways. For example, WebDAV-compliant tools can be used by a virtual organization to develop business plans, create software or develop libraries of information. Together, WebDAV and BPML provide the basic foundation for collaborative process development.

BPML does not create processes, nor does BPML limit the ability to define processes in other languages. BPML simply allows a definition of these processes to be exchanged between software tools and provides a basis for the development of open process systems. BPML is an implementation-neutral interchange format for expressing processes, much as EDI is an implementation-neutral format for expressing business transactions and as HTML is an implementation-neutral format for expressing the presentation of Web pages. And just like HTML, business users are highly unlikely to work with BPML directly, but instead will use a variety of tools that rely on BPML as a common language,

much as Web authoring tools such as Adobe's Dreamweaver, Microsoft's FrontPage and countless others allow people to develop and maintain Web sites without needing an in-depth knowledge of HTML.

BPML makes all processes explicit by representing them separately from the software infrastructure in which they reside. Similar developments have occurred before in the history of computing. Operating systems dramatically simplified software application development by removing machine-language-level considerations from application development. Likewise, many corporations manage their business rules in a separate business rules management system (BRMS) rather than embedding rules in each and every application. Much as the database removed responsibility from the application for the precise arrangement of data, the BPMS and its BPML foundation remove responsibility for the management of processes from process components. Extending this "separation of concern" to business processes means freeing information system developers from largely clerical, machine-oriented tasks, so that they can take on the more interesting and rewarding challenges of enterprise process design.

Tomorrow's Process Landscape

Every CEO, CIO, CFO, supply chain director and management consultant on the planet has imagined business models and the associated business processes that they would like to implement, right away. They are not short on new ways of imagining how to improve their businesses. Until the third wave every new business process has been hard to achieve in practice due to the cost, time and technical effort needed to implement the required software. BPML has eradicated that difficulty.

Before the BPM innovation, putting new business models into operation depended on the use of complex distributed computing, messaging and integration solutions. With the advent of BPML, these complex and expensive solutions will become simpler, less expensive and ultimately disappear from the mind's eye of the information systems developer. How many young database developers know the details of logical and physical input/output control systems (LIOCS and PIOCS)? They are still there, but the DBMS and high-level languages such as SQL push them down into the plumbing—out of sight, out of mind. Like-

wise, the complex plumbing of distributed systems can now be managed by the new process technologies that offer simpler, more cost effective and manageable alternatives.

Further radical simplifying steps will no doubt be taken as the process era unfolds. Some companies evaluating BPML have already noted that it is possible to model and execute value chain-wide business processes on a single centralized computer system. The *design* of the end-to-end business is fully "distributed," but its *execution* takes place on a single system. All the participants in the supply chain are present —suppliers, manufacturers, distributors, customers—but the technical implementation and *its management* can be centralized. The process-center, like the data-center, will soon be commonplace. Even in such a centralized environment, the business-level participants still exchange "messages" with each other—they still maintain their own "state," their own process data available to query. Transactions, business ones that is, can still be unwound.

The centralized execution of massively distributed business level process designs is going to be something many organizations will want to explore, especially if they want to incorporate the "little big men" in the value chain. In the U.S. these are the millions of small and medium size enterprises (SMEs) that can be the weak links in any value chain. They are the suppliers' suppliers and typically employ 50 or fewer employees and perhaps have nothing more that an Internet connection and spreadsheets for business technology. Yet they are the backbone of the American economy. A single, multi-company, business process management system, can be responsible for coordinating the activities defined by the end-to-end process (sequencing, synchronization and scheduling); managing process instances (lifecycle, persistence); and processing distributed transactions (two-phase commit protocol, open nested transactions and compensating transactions)—things the SME never imagined, but the very things an industry's 800 pound gorilla can extend to its suppliers' suppliers' suppliers to the benefit of all.

Like data management systems, the business process management system will, over time, move into the background becoming part of the technology infrastructure taken for granted in every company. The innovations built on data management have been truly mind boggling—an infinite variety of data-aware applications. The applications built on BPM will be equally surprising—an infinite variety of process-aware

applications. But here and now, one immediate advantage of BPM is the ability to align process design with organizational objectives, a first step in understanding the wider implications.

BPML cannot be used to model the high level values and goals of the business—these activities will remain forever in the domain of human insight, creativity and intelligence. BPML can, however, play a key role in helping identify whether goals are being met. For example, queries against a set of process instances can build a view (past and present) of whether goals are likely to be met. It is also entirely possible to query BPML to determine which participants (systems, users or processes) are responsible for achieving goals or what might cause a goal to be unreachable. BPML is the foundation for a process level of business intelligence.

Putting It All Together

As the medium for the convergence toward process-oriented enterprise computing, BPML provides interoperability between applications, process management systems and a host of new process tools. BPML was designed for complete business process management —discovery, design, operation, optimization and analysis. It is the foundation upon which both software companies and businesses can develop the next generation of process-aware systems, tools and applications.

Business processes have heretofore been second-class citizens in IT because of the challenges in developing a representation, and execution environment, able to cope with the dynamic, expanding, contracting, changing activities of the business. Founded on process calculus and the computer science of mobile processes, however, BPML is designed to manage this complexity and dynamism so that business processes can become first-class citizens again.

The theoretical basis for BPML, like all formalisms, is complex. The same is true for the deceptively simple data model where the underlying relational algebra provides confidence in the safe storage and utilization of the business assets managed by the database system. Ditto process management systems build on the foundation of process calculus. Fortunately software developers and business users do not work directly with the formalisms. They use high-level tools and many are unaware of the formal foundations on which those tools were built.

Languages being developed such as the Business Process Query Language (BPQL) enable the gathering of metrics and decision support information for all of the industrial and enterprise processes.

Using the first principles of architecture, the design of BPML is a balance of goals and constraints meant to fit its purpose. It couldn't be so complex that software vendors couldn't implement it, nor could it be trivial. The first principles underlying process management systems had to be applicable to a wide range of purposes. BPML was designed using this bottom up approach—from theory to implementation—so that all higher-level languages and systems built on it would inherit the benefits of its strong foundation. Historically, progress in the development of computer systems has been accomplished in this way, beginning with the foundation of the binary numbering system chosen by Dr. John Vincent Atanasoff when he built the first electronic digital computer during 1937 – 1942 with Clifford Berry at Iowa State University. The binary numbering system continues to serve as the formal underpinning of today's computers, yet computer users couldn't care less. Today, Pi-calculus has been chosen as the foundation for the new "business computer." In the future, business people won't care less.

Few worthwhile endeavors are easy, and BPML is no exception. Moving the workflow document and the application interface out of the center of the computing universe to focus on processes is a paradigm shift for software companies, and for any business, but a necessary shift. Some have said that BPML and all that goes with it exceeds current needs. But for those who have implemented significant integration projects using previous technologies, the old proprietary and point-to-point solutions are an even more daunting prospect. Since companies can almost never be exactly on the curve, their choice is to be behind it or ahead of it. Being ahead of the curve is so much easier in this case.

References

[1] Arkin, Assaf, *Business Process Modeling Language (BPML) Specification*, BPMI.org (www.bpmi.org)

About the Authors

HOWARD SMITH is Chief Technology Officer (Europe) of Computer Sciences Corporation (CSC) and co-chair of the Business Process Management Initiative (BPMI.org). With more than 24 years in the IT industry, he is a sought after speaker and advisor. His work in predicting and shaping technology at the intersection with business led him to take an active role in the development and application of the third wave. He is currently researching the application of *business process management* to corporate sustainability, innovation and growth, for which he has global research and development responsibility at CSC.

PETER FINGAR is an Executive Partner with the digital strategy firm, the Greystone Group. He delivers keynotes world wide and is author of the best-selling books, *The Death of "e" and the Birth of the Real New Economy* and *Enterprise E-Commerce.* Over his 30-year career he has taught graduate and undergraduate computing studies and held management, technical and consulting positions with GTE Data Services, Saudi Aramco, the Technical Resource Connection division of Perot Systems and IBM Global Services, as well as serving as CIO for the University of Tampa.

Other Advanced Business-Technology Books from Meghan-Kiffer Press

Business Process Management: New Directions
The Evolution and Applications of Business Process Management

The Death of 'e' and the Birth of the Real New Economy
Business Models, Technologies and Strategies for the 21st Century

Enterprise E-Commerce
The Breakthrough for Business-to-Business Commerce

The Real-Time Enterprise
Powering Profits with Process Automation

MK
Meghan-Kiffer Press
Tampa, Florida, USA, **www.mkpress.com**
Advanced Business-Technology Books for Competitive Advantage