Don't Bridge the Business-IT Divide: Obliterate It!

By Howard Smith and Peter Fingar

To business people, it seems that technology is always getting more complex. Technical people feel the same way. Over the last five years, delivering business applications has become much more complex, with layer upon layer of new infrastructure requirements and new features. While this has been good for IT industry players that sell new products for new layers in the "technology stack," it isn't necessarily so good for companies that use them as business tools.

When complexity mounts and eventually becomes unmanageable, it's time for action. As Walt Disney once said, objecting to a proposed sequel to his "Three Little Pigs" cartoon, "You can't top pigs with pigs." In the world of business, stacking a thousand doghouses, one atop the other, to build a skyscraper is a great proposition for doghouse vendors, but not for future occupants. Skyscrapers need an architecture of their own — their own paradigm, not a sequel to the doghouse paradigm. Business processes need an architecture of their own — their own paradigm, not a sequel to the workflow, EAI, messaging, or Web services paradigms.

The spreadsheet is a simple yet eloquent example of a useful paradigm shift. The con-

venience and low cost of the breakthrough was so striking that it led to the PC revolution in business. The spreadsheet could not have been successful had it not been for the fact that PCs, a standards-based commodity, were spreading like wildfire. To the business, the PC loaded with a spreadsheet meant a radical simplification of routine calculations, transferring to the everyday business person a function that had once required special programming skills.

A similar simplification and transfer of functions is needed by those pursuing business process development and optimization, for as the management prophets foretell, the next phase of corporate development will require systematic control of the value chain, rather than narrow-gauge process fixes. Michael Hammer has admitted that managing such wholesale change is mind-numbingly complex. In fact, it's no longer possible without computer assistance. The technology-planning horizon for Global 5000 companies is now a synthesis of software engineering and process engineering. With the widespread adoption of application servers, component-based development and Web services, the field is ripe for the spread of process management.

Some paradigm shifts the IT industry created have been truly "radical" — disruptive, costly, and unappealing. The third wave of Business Process Management (BPM) is different. The architects of the third wave bowed low to the futility of trying to persuade business to switch from one three-letter acronym to another. They paid close attention to the urgent needs of businesses to preserve, extend, and flex their existing investments, as well as to enable future opportunity and growth. Building on what already exists in companies everywhere, the third-wave innovators have provided a new level of convenience — the open vista of native process management: heritage-friendly but legacy-free. Those seeking a sound business and technology architecture for building the process-managed enterprise will think twice before stacking doghouses and then calling the result "The Company of the Future."

The message is equally clear for technicians, both in the software industry and the IT shops of major corporations: build new applications on a process foundation, for the CxO team expects nothing less. Embrace process management in the way companies adopted data management decades ago, by separating out data for application-independent management, analysis, and controlled sharing. At that time, companies knew they had a data problem, and they responded by recognizing the value of relational data management systems. We believe that companies are now recognizing they have an analogous process problem. The balance of power in the business-IT relationship must shift away from the need to squeeze business processes into the prepackaged application "fashions" of the IT industry and toward the ability to design, improve, and transform business processes that BPM enables.

The third wave of BPM does more than facilitate process design. It provides a direct path from vision to execution. It's not so much a matter of "rapid application development" as "remove application development" from the business cycle. Show the BPM capability to any executive, at any level, and they'll understand inside five minutes how to break through the IT logjam. Some may still want to prevent managers from defining business processes themselves, saying it's too complex a job and should be left to specialists. That may be true right now, but it won't be by the week after next.

Changing the Change Process

For those managing change, it often seems there's a sharp distinction between the process of change and the change being introduced, but this is an illusion.

Consider the words of David P. Norton, ex-CEO of the consulting firm, The Nolan Norton Institute, and now a director with the Balanced Scorecard Collaborative: "To execute strategy is to execute change at all levels of an organization," he said. "Overlooking this truth is one of the greatest causes of a failed transformation effort."

Whatever we choose to call the "change process," it changes over time. Those changes need to be managed, just as it exerts control over the processes it seeks to improve or introduce. Any theory of process management must recognize this and break down distinctions between the process of change, the process under change, and change in both.

Change occurs for many reasons. Examples include changes in business regulations, audit requirements, or improvements in our understanding of how to mobilize and catalyze resources. Distinctions between different types of change only arise as a result of today's inadequate, disparate, and disjointed process management methods and technology systems. In the third wave, there's no need for a distinction between the change process (e.g., Six Sigma, project plan) and the process under change (e.g., customer service, product manufacturing). They can be one and the same.

Third-wave processes are inherently open to change; theorists call this "mobile" behavior. Mobile systems are systems whose participants freely communicate and change their structure. They do this in two ways:

- Links between participants, which represent the relationships between them, may themselves change. New links may be formed, old links being broken. To an observer this looks as though the participants, or the links, are moving.
- Another type of mobility is the relationship of the linked participants to the environment in which they exist. This, in turn, looks as though the whole process, or subset of linked participants, is moving. Such "mobile" processes now pervade the informational world of computer systems and networks as well as the wider real world of which they're a part. Examples include industrial supply chains, the Internet, cellular mobile telephony networks, air traffic control, and distributed computing.

In a supply chain, a message may indicate the physical movement of goods or the level of inventory in a warehouse. On the Internet, protocols between routers establish preferred routes for packet switching. In a mobile telephony network, a base station announces the availability of circuits to receivers in the same area. In air traffic control, aircraft move into and out of controlled airspace. In a corporate IT infrastructure, messages create relationships and share data between different business applications. Communication and interaction within such systems is rich and varied. Modeling such behavior is hard, but crucial if effective process management tools are to be developed.

At every level, there's change. What we commonly refer to as a "business process" is no different. We now understand that trying to capture these descriptions at a

specific instant in time, in a software application, is futile. We need a more mutable digital form: prone to change, capable of change and of being changed.

Participants in a business process include employees, information sources, business units, computer systems, business partners, machines, trucks, goods, even business processes themselves (for example, as occurs in outsourcing). Change occurs through the acquisition or loss of these participants, through the growth or contraction of relationships among them, and their interactions with the environment. A business process "moves" — as it changes — in the multi-dimensional space of time and structural evolution. Like a living entity under the influence of Darwinian evolution, it exists in the past and the present, and has possible futures.

In understanding a business process, we therefore distinguish different characteristics such as state, capability, and design:

- State can be understood through the execution of the process, the values of calculations performed, and the information collected and generated along the way.
- Capability can be understood as the specific participants present within the process at any time: what they are, the activities they're capable of performing, and the relationships established among them and through which they communicate.
- Design can be equated to the intentional characteristics of the process — those put in place during the design process, before the process was set free to execute, to evolve, and to change. Thus, we speak of the difference between intentional design and observed behavior.

How can we start to link the world of business management with the world of business technology using these process characteristics? Perhaps we can think of operations, the everyday running of the business, as being mostly concerned with process state. Perhaps business strategy is largely a matter of intentional process design. Somewhere between the two lies the management of process capability, the actions taken to grow the business, its customers and its market share. If we accept these rough analogies, then:

- Business intelligence is an analysis of the past and present state, capability, and design of business process.
- Business insight is a simulation of future state, capability, or design.

Today, while the analysis and simulation of process state is perfectly well understood, what does it mean to analyze and simulate growth in a process's capability, or changes in its design? The third wave points to possible answers.

Computing's New First-Class Citizen

In the esoteric world of technology, there are plenty of ways to develop IT applications that automate business processes. The market has a nasty habit of regularly generating new technologies that tease developers into trying out new ways to develop computer software. Not only do new computer languages constantly appear, but new methodologies, each with their own "first-class citizen," the center of their computing universe. Recent examples include peer-to-peer, grid and utility computing, self-organizing data networks, the semantic Web and distributed agents. This constant evolution in computing infrastructure, information types, applications, and process approaches is confusing to business people. They wish to find technology that reflects the way they understand their business.

When technologists examine the applicability of each of these new techniques, and compare them with more established practices such as component and object-based software development, they are, in effect, examining their relevance to modeling and support for the classical six domains of change within a company. The domains are: process, organization, location, data, application, and technology.

Yet process isn't a category. Process encompasses change in the representation of the other five. Therefore, in developing a process representation language, the third-wave innovators looked not for a single new theory, but a theory that provided a synthesis of other theories. In process calculi, they found approaches that could describe the previously separate descriptions of a company's organization structure, locations of operation, data model, application logic, and technology infrastructure requirements. They then asked, "What if this were the basis of a new foundation for computing?"

Just as there are countless ways for technicians to implement an automated business process, there are just as many ways to develop and execute a business strategy.

Just as there are countless ways for technicians to implement an automated business process, there are just as many ways to develop and execute a business strategy. In *The MacRoscope*, Joel de Rosnay postulates, "The fundamental concepts that recur most often in biological, ecological, and economic models can easily be grouped into major categories: energy and its use; flows, cycles, and stocks; communication networks; catalysts and transforming agents; the readjustment of equilibriums; stability, growth, and evolution. Above all, the concept of the system — living system, economic system, ecosystem — that binds all together all the others."

Similar thinking can be applied to existing management theories, including Six Sigma, Total Quality Management (TQM), balanced scorecard, activity-based costing, Economic Value-Added (EVA), and value analysis. Each can be represented using a small set of core concepts — an ontology of commerce — a vocabulary of well-defined primitive forms that can be combined in a recursive procedure to represent any business process, no matter how complex and how expansive.

Examples of such primitives include ideas such as identity, relationship, exchange of value, and knowledge about

a system. We now understand that the outcome of a process is nothing more than the result of the flow of interactions among participants at every level, from the digits involved in a calculation to the most complex exchange of assets between two economic entities. Formalizing this business vocabulary and providing precise definitions of these elements will emerge as the counterpoint to the exploitation of process management systems over the coming decade.

Unification of Data, Computation, and Interaction

In process calculus, relationships represent anything from a physical link (a truck arriving at a warehouse) to a business relationship (two parties entering into a contract) to a mathematical property (such as calculating tax). Using just a tiny set of primitives, these theories can unify both the large-scale (macro) structure of mobile process systems and the small-scale (micro) structure representing intricate behaviors, themselves processes.

Previous theories in computer science, notably the Lambda-calculus, focused on the behavior of much simpler computer systems, where there's either a single thread of execution or a set of parallel but non-interacting tasks. Such algorithms are procedural, sequential, goal-oriented, hierarchical, and deterministic.

All of today's well-known programming languages can be studied using Lambda-calculus, including FORTRAN, COBOL, Pascal, Lisp, C, and Java. By contrast, in process theories such as the Pi-calculus, the main focus is on systems that interact and interrupt one another, where there are many deeply nested, independent, but coordinated, interacting threads of execution. Business processes are an example. The differences between these theories are striking, for even our notion of what constitutes a common-sense interpretation of data and value has utterly changed.

In conventional computer languages, there exists the concept of a "type." For example, the type of the number "five" is called an integer, the type of the text "hello" is called a string. Such types represent values — such as 24, "customer name," "purchase order number." These values are then aggregated to form records and these are stored in databases. All conventional languages focus upon computation using values and records; for example, counting customers that match certain criteria or evaluating their credit worthiness.

By contrast, in languages derived from Pi-calculus, types represent behavioral patterns. In business terms this would mean things such as "signing a new customer," "exchanging contracts," or "performing work." Here, computation is harder to envision. To do so, think of analogies such as "measuring the acquisition cost of new customers," "understanding the value exchanged through a negotiation," or "analyzing work habits." If brown is the new black in fashion, then behavior is the new data in the third wave.

Process engineers and designers of process management systems respect Pi-calculus because it identifies the concepts that underpin a wide variety of concurrent systems. This is no different from other fields of study. Electrical engineers respect differential calculus. For them, it ties together various frameworks, concepts and thinking without distortion, and defines what is common to all electrical systems. Likewise, database engineers respect the relational data model originally proposed and defined by E.F. Codd.

Process calculus would, however, be of no use if it were wonderful at describing business processes but omitted to describe traditional computational processes. Fortunately, it can do both. For example, the operation to combine two lists into one — a common operation in most computer programs — can be regarded as a change in the relationship between the lists, from disconnected to connected. In process calculus, the movement of a piece of data inside a computer program is treated exactly the same as the transfer of a message, or indeed an entire computer program, across the Internet. Taking decisions and computing results — in fact, all common programming tasks — can be represented and understood as "processes."

This process approach, where process is the new first-class entity, can be applied even to the lowly task of adding two numbers. The sum itself is a process. It can be understood as a process in two ways. The first is: The "one" participant joins "two" in the "plus" relationship, which then grows to include the participant "three" in the "equals" relationship. The second is: A participant receives a message containing "one" from another participant, and a similar message containing "two" from a second participant. It then performs the "plus" activity and sends the result "three" in a message to a third participant.

These rough analogies hardly give credit to the richness of Pi-calculus, but they help us understand how any calculation can be represented as a process. This perspective gives the third wave of process management its inherent ability to capture, describe, and manage whole processes — not just integration between existing algorithmic procedures written in conventional software languages and embodied in today's packaged software.

This approach to process representation can be applied to a wide range of problems. Other examples include:

- HyperText links that are passed around, created, or which disappear
- Connections between mobile telephones and base stations
- A job scheduler, allocating work on a production line
- References passed as arguments to methods in business objects
- Business partners changing roles in a business process
- Work passing between participants in a business process
- Code sent over a computer network to execute on another system
- A vending machine, serving up a soft drink
- A mobile device acquiring a new capability over a telephony network
- Procedures passed as arguments to methods in computer systems
- A business process passed to a business partner for execution.

Relationships, specifically their names, appear to be something fundamental. They're ubiquitous in computing, in the form of addresses, identifiers, links, pointers, and references. They're ubiquitous in business, in the form of purchase order numbers, product codes, organizational roles, the identity of partners, the types of business relationships and the infinite ways of organizing, performing and referring to items and patterns of work.

In process calculus, the names of relationships can be used for many purposes, such as:

- Channels over which processes communicate
- The names of processes themselves
- The names of objects (as in tradition-

al programming)
- Proxies for physical locations
- References to encryption keys.

Names received in one interaction can be used to participate in another. By receiving a name, a process acquires a capability to interact with processes that were previously unknown to it. A name can even refer to the specific pattern of a process design. The connections among participants — the structure of the system — can thus change over time and in ways the process designer didn't envision.

Process-Aware Applications

In business, a supply chain director is required to understand complex processes, such as a logistics network, but his or her ability to manage and optimize that network is still an art, rather than a science, and presents huge challenges. By contrast, to a software engineer, even the most simple business process, such as the exchange of a purchase order, can be difficult to program and hard to represent. This is because today, most software engineers use constructs such as records, objects and interfaces — first-class entities of a past era, not attuned to the representation of business processes. This mismatch between the reality of business and the artifacts of traditional software limits the ability of the software engineer to provide tools to the supply chain manager to help manage real business.

In many industries, business and systems architects strive to create software applications that accurately reflect their business. Sometimes they don't realize that a perfect simulation is their ultimate aim. Architects in other industries know precisely that this is their task. In the logistics industry, companies often model their IT architecture closely around the behavior of the physical logistics networks they monitor and control. The tools they have to achieve this are improving all the time.

Gradually, architects are finding ways to represent the behavior of complex systems — interconnected and inter-related mobile processes — within the business applications they develop. Soon they'll realize that mapping business concepts into artificial IT artifacts such as objects, interfaces and procedure calls, should be replaced, or at least complemented, by the process calculus models of the third wave. These artificial constructs arose to support the composition of software, not the representation of business. Architects are now looking for methods, tools, and systems that are purpose-built for business. Increasingly, they're looking to business process modeling languages for solutions.

Drawing on themes from concurrent programming and agent-based systems, process-modeling languages treat process participants as autonomous "agents." An agent is free to act, having both an internal process that it follows, and an external interface through which it communicates. As it interacts with other agents, it collects information from them that it can refer to in making decisions, calculating results, and constructing messages to send to others. It can then pass on this derived information to other participants.

Partly autonomous, and partly constrained through process design, agents act in parallel and in concert. The action of any particular agent depends upon both the anticipated actions of the agents with which it interacts or the agent's analysis of the state and structure of the system process in which it exists. Such systems are called "adaptive complex networks," and they occur widely in nature, not just in computing. Economic networks and value chains are examples.

Imagine a typical business application today and ask: "Why does it stay the same? It's digital stuff, right? So why can't it change? Why does a new version of the application have to be developed for every situation? Why can't it adapt to me?" It's now legitimate to ask such questions. IT industry experts are already anticipating software programs that will write themselves to agreed process patterns. IT infrastructures will take on — auto-magically — the form of the organizations that use them.

From now on, the business process is the "app," and the "app" is nothing more than mutable data. For this is the "third wave" form of business asset. Let's move on and build the new "process-aware" applications of the third-wave and not try to preserve paradigms that fail to fully represent the complexity of business. Those companies wishing to take this step will need a strategic infrastructure designed for the purpose, the BPM system upon which they can manage their "mutable process data."

Conclusion

In conclusion, it might be well to recall what Peter Drucker once said about predictions:

"It isn't so difficult to predict the future. It's only pointless. But equally important, one cannot make a decision for the future. Decisions are commitments to action. And actions are always in the present, and in the present only. But actions in the present are also the one and only way to make the future."

Winning companies will invent, not forecast, their own futures. BPM provides the ability to create the future by innovating with process, without the costs of reengineering or adding more layers to the already complex technology stack. A business experiment that fails is a business that failed to experiment. The third wave of BPM is new, and it needs to be done — now.

This article is excerpted from the book, Business Process Management: The Third Wave, *published by Meghan-Kiffer Press (ISBN 0929652339).*

About the Authors

Howard Smith is chief technology officer (Europe) of Computer Sciences Corp. (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. Website: www.bpm3.com

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. Website: www.bpm3.com*