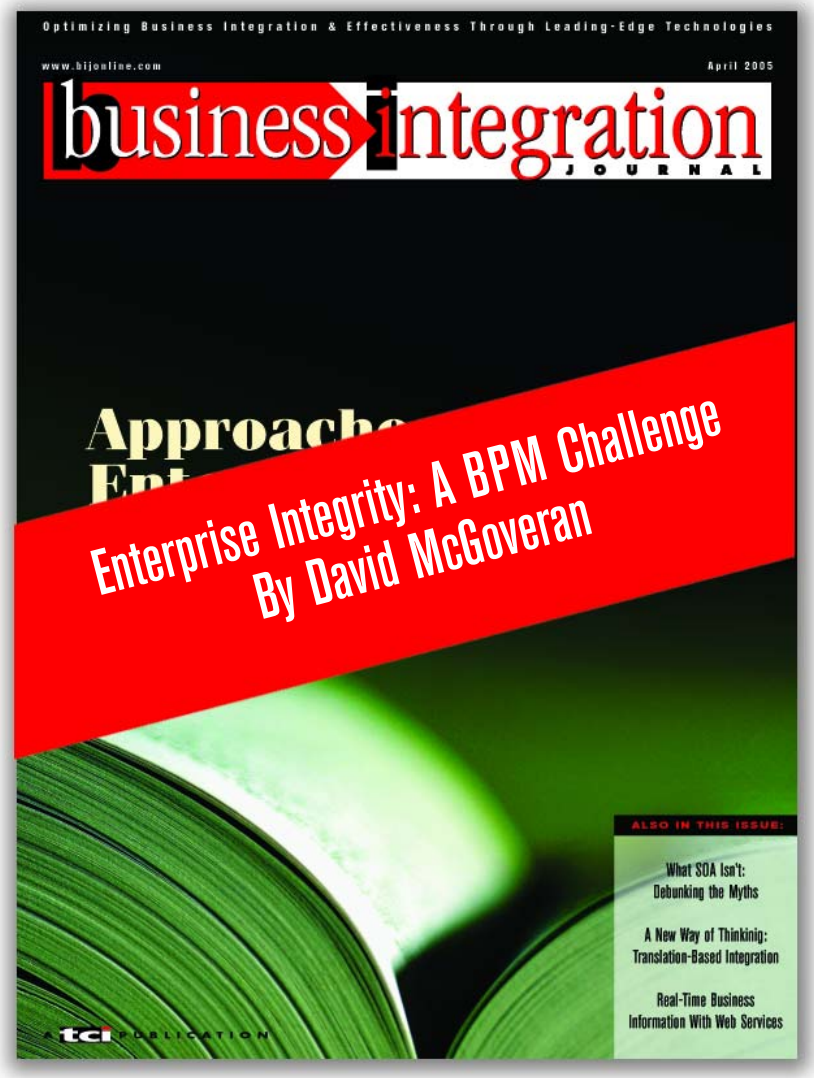
This article appeared in the
April 2005 issue of

business integration JOURNAL



Subscribe instantly at
www.bijonline.com

- Free in the U.S.
- $48 per year in Canada and Mexico
- $96 per year everywhere else

ENTERPRISE INTEGRITY

A BPM Challenge

BY DAVID McGOVERAN

Massive marketing efforts have gone into concepts relating to business process, including Business Process Management (BPM), composite applications, and Service-Oriented Architecture (SOA). The pages of *Business Integration Journal* have been filled with such articles and advertisements. In fact, it seems like everywhere you look—from the industry press, conferences, and standards specifications, to vendor literature—you find references to business process. Various technical analogs have appeared on the scene, including services orchestration and choreography. These technologies are still far from mature and so, as an analyst, I've found that developing comparative product evaluations that cover product functional variances, while being honest and useful, can be quite frustrating. It's easier to challenge an entire industry's efforts.

I first wrote in these pages about the challenge and potential of BPM in 1999 and then an entire series starting in 2001. Perhaps the most valuable and most ignored BPM System (BPMS) characteristic described therein was that of resource independence. We can define a BPMS, somewhat loosely, as software that enables business process execution from a Business Process (BP) model understandable to and specifiable by business analysts, with closed loop monitoring and continuous process optimization. A BPMS exposes the business logic and business rules that define the business process, making them far easier to understand, create, maintain, and optimize than when they are embedded in application code. Of necessity, business processes comprise a wide variety of business activities implemented by some combination of human, software (e.g., legacy transactions, database stored procedures, monolithic applications, reusable services, etc.), or machine resources.

Having the resource independence means that the specific method of accomplishing a desired business activity isn't embedded in the business process model. Business managers shouldn't have to worry about which resources are used, so long as requirements (including performance, quality, and cost) are met. Instead, the business requirements of a business activity logically determine which available resources can accomplish the task in such a way as to meet business objectives. Resource independence has many implications (more than I've space for here), and its benefits can't be overstated. Many products force user awareness of resource assignments at model definition time. The modeler may be forced to make explicit assignment of a specific resource (such as a Web Service vs. a manual worklist). Many products simply exclude entire classes of resource, supporting only manual or Web Services implementations, and are better characterized as workflow or process automation products. Even modeling diagram icons may depend on the assignment. Resource dependence can also be implicit and therefore more insidious, with the chosen type of resource placing restrictions on permissible process definitions. Resource independence demands certain facilities be cleanly separated: business activity requirements specification, resource capability specification, deferred assignment, and run-time scheduling.

Consider some of the benefits of resource independence: Perhaps most important, it helps isolate business decisions from the ever-changing world of IT and human resource management, letting business process modelers concentrate on business issues. Consider a business process comprised of manual activities. With resource independence, process definitions need not be changed as activities are selectively automated if the automated implementation meets or exceeds manual capabilities. Suppose a critical software implementation fails. With resource independence, an alternative manual procedure can be implemented transparently. As yet another benefit, such a BPMS or orchestration product can more easily accommodate decisions to shift responsibilities for business activities among trading partners. And, with run-time resource assignment and scheduling, business managers can change resource optimization rules: from production volume to quality to cost efficiency.

No BPMS product meets all the requirements presented in the 2001 BPMS series. That's understandable with developing technologies. More disappointing is that very few BPM products, let alone products that purport to enable BP integration, BP automation, BP optimization, services orchestration, services choreography, or services composition offer any degree of resource independence at all. After six years, you would think someone would get this right.

And so, I issue a challenge, not only to BPMS vendors, but to all software vendors that say they provide some form of process, orchestration, or choreography: Be the first vendor to support resource independence. If your product has been as well-architected as you say, and if you've really understood the business needs of your customers, you should be able to do this by the next major release. It's time to meet the challenges of delivering truly business-oriented software. Resource independence would be a good start. The *integrity*, efficiency, and continuity of your customers' *enterprises* demand it. **bij**

About the Author

David McGoveran is president of Alternative Technologies. He has more than 25 years of experience with mission-critical applications and has authored numerous technical articles on application integration.
e-Mail: mcgoveran@bijonline.com
Website: www.alternativetech.com