enterprise integrity

By DAVID McGOVERAN

BPMS Concepts, Part 5

In two recent columns, we examined the technical components (or kernel) of a Business Process Management System (BPMS) and business interfaces to the kernel. We've not yet considered a few remaining components in a BPMS. Without them, a BPMS is little more than a planning tool. For now, we'll refer to these components as technology interfaces because they enable the BPMS to interact with other systems.

The following technology interfaces can be either simple or sophisticated, but some version of them must exist if the BPMS is to manage real business processes and activities. Two of these are considered "optional" because they're not part of the BPMS per se, as explained below.

- **Interface manager** — A BPMS is of little value if the process engine cannot communicate with business functions as implemented. It must be able to communicate both control flow and data flow in a coordinated fashion, though these may be separately defined and quite distinct. This is far from trivial. Few interfaces are designed for anything other than data flow! If the BPMS is integrated with a suite of business integration components, it's this BPMS component that's responsible for the operational aspects of that integration. The interface manager handles support for transports and adapters (whether to middleware, applications, or presentation software).
- **Activity manager** — The activity manager is closely related to the interface manager. Activity management is the orchestration of all the Enterprise Application Integration (EAI), business-to-business (B2B), and business-to-consumer (B2C) infrastructure, including people and software, via the interface manager. This encompasses intelligent adapters, the adapter Software Development Kit (SDK), messaging and other transports, data transformation, and all the usual "broker" technologies. For example, if a data transformation or personalization is needed between business functions, the activity manager is responsible for recognizing this and invoking appropriate components through the interface manager. A caveat to remember is that vendors are starting to position their products as BPMS solutions when some such products actually offer little more than activity management.
- **System manager** — A BPMS requires a facility for installation, configuration, and system management of its components. This isn't a user's facility, but an IT support facility. The system manager component of a BPMS should have all the usual desirable properties of an enterprise-class software system manager or administration component. A system administrator's job is difficult enough without adding complexity here, so usability and reliability are paramount. The goal is elimination of manual administrative tasks, "error-proofing," and online guidance. Don't be surprised if early BPMS products require multiple and even semi-manual component installation. Making certain that the platform and environment are complete and properly configured may be quite tedious in first-generation products.
- **Integration components (optional)** — At one end of a spectrum, integration components are a set of adapters that provide point-to-point integration between the BPMS and means used to implement business functions or activities. At a minimum, a BPMS requires an adapter to communicate with people for manually implemented business functions. Certainly, there are many business needs a simple BPMS with one such adapter could solve. At the other end of the spectrum, integration components may be a full suite of business integration components and services. We'll discuss this further later, since it connects BPMS with a standard EAI stack. Clearly, a BPMS best operates in the context of a complete suite of business integration components and services.
- **Integrated Development Environment (IDE) (optional)** — Finally, as BPMS usage matures, users will undoubtedly want to develop applications that best take advantage of BPMS capabilities. To that end, a suite of development tools is needed. In its simplest form, such an IDE enables the development of new adapters that are process-aware. With more sophistication, an IDE for designing and developing process-enabled, event-driven, and rule-based applications or application components is highly desirable. To my knowledge, such an IDE does not yet exist, although one can cobble it together with a collection of existing products. Either way, there's a new integrated process-object methodology that individuals and organizations should understand before marketing or using such tools.

The integration of these components into a single system dictates many of their features and functions, which I won't elaborate on here. It's sufficient to say that many first-generation BPMSes will err in the rush to market. Through a future column, we'll explore these components from a slightly higher, conceptual view. Until then, may all your integration be enterprising. eAI

David McGoveran is president of Alternative Technologies, Inc. He has more than 20 years' experience with mission-critical applications and has authored numerous technical articles on application integration. e-Mail: mcgoveran@alternativetech.com.