

# enterprise integrity



By DAVID MCGOVERAN

## B2B Success Secrets, Part 2

I've written in the past about business transactions, and how they relate to computer transactions. To summarize, physical transactions are units of recovery, logical transactions are units of consistency, consisting of physical transactions, and business transactions are units of audit, consisting of logical transactions. In other words, the beginning and end of a business transaction are auditable, consistent, recovery points. Business transactions can be inordinately complex and highly variable duration (ranging from microseconds to decades). Managing business transactions successfully with other business entities is, of course, critical to the success of any B2B operation. But managing them is a daunting task.

Early in my career, computer transactions were as difficult to imagine managing in some consistent way, let alone automatically, as business transactions are today. Essentially all transactions were implemented programmatically and uniquely for each transaction (even those in the same application). With the advent of multi-user systems, refined transaction models, and early DBMSes, it became possible to provide transaction services in the form of TP (transaction processing) services via monitors and managers. These facilities provide crucial services, including multi-threading, load balancing, queuing, logging, coordinated recovery, all in the process of delivering atomicity, consistency, isolation, and durability. When multiple, independently logged and recovered computing resources are involved in a transaction, a distributed transaction manager is required.


Just as distributed computer transactions span multiple independently managed computing resources, business transactions often span multiple independently managed business resources. However, despite the similarities and the relationships to computer transactions, treating a business transaction as though it were just a type of computer transaction is a common mistake made by computer scientists and IT professionals. Business transaction initiation, completion, and abandonment are controlled by external business events, not by a deterministic program. Much of the business process from which a business transaction derives its proper context consists of coupled, human-controlled activities that are strictly asynchronous and, therefore, result in unpredictable event schedules and behavior.

The behavior current business transactions systems produce is reminiscent of that which results when multiple transaction processing programs, each with their own model of a transaction, interact around common resources. The problem of unpredictable behavior was solved by imposing a common transaction model, typically through a transaction manager and TP services. This suggests that the industry needs a business transac-

tion manager and a standard business transaction model.

What would a business transaction manager do? Well, just as multi-threading through a transaction manager permits multiple users to perform an isolated task while sharing the most resources, a business transaction manager permits multiple business entities to perform an isolated business activity while sharing the most business resources. Private business processes are analogous to threads. Public portions of that business process correspond to synchronization events around global, shared resources. A business transaction manager would guarantee that the business transaction is atomic across business entities and that the business rules jointly imposed by the involved business entities were respected (consistency). The business transaction property analogous to isolation is security, and guarantees that only business transaction participants can gain access to common resources or results. Of course, private resources and results can be promoted to common, shared status only by their owners. The property corresponding to durability, auditability, should be implemented as non-repudiation.

Obviously, I don't have space to describe a complete business transaction model. However, it should be clear that such a model is possible. In fact, there are too many models possible, most of which work only in a limited context. And if a business transaction model is possible, then a business transaction manager is also possible. Likewise, this is not the place to describe a design in detail. In many ways, B2B trading hubs implement some of the facilities of a business transaction manager, albeit somewhat haphazardly. Some vendors (such as NEON/Sybase, Extricity/Peregrine, and others) have inadvertently implemented most of the required technology.

As with transaction processing, business transaction processing application development will not become truly productive until we have a standard business transaction model and business transaction managers. And until that happens, B2B will be a discipline with lots of "one-off" implementations, requiring considerable skill to integrate. In the meantime, now that you know some of what B2Bi infrastructure should be providing, you can evaluate commercial products or even construct the facility from commercial components. Remember: Transaction management is about maintaining integrity, and enterprise integrity applies to all the business entities involved in an enterprise. 

*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](mailto:mcgoveran@alternativetech.com); Website: [www.alternativetech.com](http://www.alternativetech.com).*