

enterprise integrity



By DAVID MCGOVERAN

B2B Success Secrets, Part IV

Last month we considered a core problem for success in business-to-business (B2B) environments and in all businesses: designing in robustness. To put it another way, what makes a business operationally “non-stop?” Making a business robust at the expense of efficiency is easy but unacceptable. This article explores some key issues surrounding this important topic. The key is to understand the distributed system nature of businesses.

The IT industry knows that software systems can be made more robust through redundancy. Even if software is monolithic, introducing fail-over to a hot standby creates a rudimentary distributed system. By introducing redundant subsystems, we improve the chances that at least one subsystem is always available. The same is true of any system, even a business.

Problems arise when we try to manage redundancy so that all the redundant subsystems are peers (there’s no presumption that one is initially primary) and yet overall efficiency isn’t significantly decreased. We must keep all the redundant subsystems “hot,” and be able to detect any failure, and transfer control to the appropriate subsystem at any time. This adds measurable overhead. Worse, additional redundancy doesn’t improve availability linearly. It’s a simple case of diminishing returns.

We can attack the diminishing returns problem if we approach the availability problem differently. Instead of thinking of redundant systems as parts replacements, think of them as ways to divide the work so more than one subsystem is working simultaneously. This makes the overall system more scalable in terms of availability, and also lets the system accomplish some tasks more rapidly.

The management problem, however, isn’t improved. We still face the problems of detecting and responding to failed subsystems, and dividing and distributing workloads (when and if the load is divisible). This extra management effort means we still reach a point of diminishing returns when adding subsystems. This time the diminished return comes in the form of performance. A failed subsystem means that the total workload must be redistributed among any healthy subsystems. Doing that may not be a simple management task!

A business must be designed for robustness, so — don’t just add arbitrary redundancy. Design problems are twofold:

- Failure coverage
- Single point of failure elimination.

Failure coverage means that each of the various modes of failure is anticipated. A management process must then be able to detect it quickly and correctly, and respond efficiently.

Business systems, like software systems, have some to many failure modes to consider. Two important modes are geographic

and functional. Geographic failure modes affect all resources in a geographic area. The solution is obviously to have a redundant subsystem that’s outside that geographic area. Taking this seriously means that the business must have the ability to rapidly transfer all operations from one geographic area to another. The remote backup site can be implemented via:

- A fully owned, fully staffed facility
- A set of outsourcing agreements
- Agreements with temporary agencies
- Agreements with competitors and strategic partners to handle workload in an emergency
- Any combination of the above.


Functional failure modes affect resources that pertain to a particular business function and are more difficult to address than geographic failure modes. Functional failures can be triggered by a variety of factors such as:

- Loss of critical supplies or communications
- Departmentwide illness
- Contaminated data
- Poorly designed business processes.

The solution is to have alternative implementations of the business function that use different resources. To be effective, such alternatives must be standard business processes.

B2B failure modes are complex, depending (via “public” processes) on other entities. Sadly, few B2B businesses have established appropriate processes to deal with either geographic or functional failure modes in this context. To be truly effective, you must not design processes to detect and respond to failures as though they were exceptions.

Processes to check for failures and confirm subsystem health must be operationally integrated into the core business. Unless such processes are responsive, no amount of B2B technical integration will help when sudden change occurs in the business environment, whether due to natural, economic, competitive, or even terrorist causes.

In a future article, we’ll discuss the second key to designing a robust, non-stop business: removing single points of failure. Until then, spread the message and remember that — technology alone won’t provide enterprise integrity. 

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; Website: www.alternativetech.com.