

business integration

JOURNAL



Reducing Risk and Protecting Investments in Service-Oriented Architectures



Reducing Risk and Protecting Investments in Service-Oriented Architectures

By Trevor Matz

An article in *The New York Times* described the behavior of a group of gambling capuchin monkeys in the lab of a Yale University researcher. A monkey first is given one grape (which monkeys covet) and then the researcher tosses a coin. Heads, they win another grape, tails, they keep the one in hand. The monkey then is given two grapes and the researcher tosses a coin. Heads, they lose one grape, tails, they keep both. The end result of both games is the same—a 50 percent chance of having either one or two grapes to eat.

Rationally (to us), it shouldn't matter which game is chosen. But when given a choice, the monkeys overwhelmingly chose to start with one grape, gambling on a potential gain instead of

starting with two grapes and gambling on a potential loss. But is this just monkey business? Does it apply to humans? More specifically, to those humans known as IT decision-makers? The answer is yes, and it's called Prospect Theory, the subject of the 2002 Nobel Prize in Economics.

This isn't to suggest that corporate IT decision-makers are a bunch of monkeys. But it does highlight some facts to keep in mind as the IT industry moves toward Service-Oriented Architectures (SOAs) and the adoption of supporting technologies such as Web Services:

- Reduction of the perceived risks is key to speeding the adoption of SOA and its associated technologies, and the realization of

its benefits. These risks include uncertainty regarding the costs and immediate benefits of SOA adoption, immaturity of the technology and related products, technology ubiquity and longevity, and applicability of SOA to the widest range of business problems.

- The faster organizations move to SOA, the quicker they'll realize the benefits of interoperability and the improved flow of information it affords within and between organizations.

Responding to Risk

IT decision-makers considering a move to SOA and adopting Web Services as an integral part of their SOA strategy typically respond to risk in one of three ways. They're:

- Waiting until the technology is more widely adopted and proven
- Testing the waters but waiting for products to mature, and for proven interoperability between standards-compliant products and platforms from different vendors, before investing fully
- Aggressively adopting Web Services and taking advantage of SOA while products mature and evolve.

Waiting to see how broadly adopted the technology will be, waiting to see if competitors are using the technology, and waiting to see if the technology delivers on the promised business benefits are time-honored responses that do reduce risk. However, while good things may come to those who wait, even better things may come to those adopters who quickly move into using the new technology. The competitive advantages they gain will be hard to beat when slower

adopters finally begin using the new technology.

Mitigating Risk

Relying on industry standards provides insulation against the risks of technology change or making the wrong technology choice. But in high-tech, the Next Big Thing always comes along, and the status quo is often short-lived. New standards and technologies invariably emerge, disrupting the existing IT environment and forcing change.

The early distributed computing standards such as Remote Procedure Call (RPC), the Distributed Computing Environment (DCE), and the Common Object Request Broker Architecture (CORBA), for example, were backed by large industry players but failed to get wide industry adoption. They were too complex, and the pervasive infrastructure necessary to support their widespread use was not yet in place. The cost of entry to the distributed computing world at that time was too high. No network effect (i.e., the increasing value of a product or service as more people use it—think of e-mail or cell phones) developed to encourage adoption of those standards.

Compared to the distributed computing example given previously, Web Services (another form of distributed computing) have a

huge advantage. The Internet, and its TCP/IP underpinning, has become the pervasive communications infrastructure for business that earlier distributed computing standards never had. The simple Web protocols (HTTP, HTML, XML) used to convey content became de facto standards well before official recognition. As a result, the Web has benefited from powerful network effects and become a self-propelled juggernaut, to the benefit of all.

An Alternate Approach

Although Web Services are critical, enabling technology for building service-oriented solutions, are they the only way? Ratified Web Services standards are maturing, and new ones are coming along to address security, reliability, and other aspects of robust enterprise applications. But at the same time, other technologies for composing and orchestrating services, some of which hold de facto standard status today, such as .NET and Java, are already ubiquitous. And still outside the scope of most SOA development platforms is the need to quickly incorporate business logic and data widely used in enterprise applications (such as in stored procedures in relational databases and other legacy technology) into a service-oriented solution without first wrap-

ping them as Web Services. This begs the question: Are Web Services always the best way to create, orchestrate, compose, and manage all the components of a service-oriented solution?

Is there a way to advance that lets IT capitalize on the promise of SOA, and the benefits of Web Services, while addressing the concern over standards and technology evolution, reducing risk, and protecting investment in de facto standards and legacy architectures? XML, the lingua franca of Web Services and the standards built on it, seems to point out the way. It provides a level of abstraction above specific technology implementations. For example, the business functionality of a service may be written in COBOL, but the rest of the world need only see its standard Web Services Description Language (WSDL) wrapper to interact with it. But is the XML-based abstraction approach the most appropriate model for data and business logic abstraction in all cases? That depends on the problem at hand and your goals.

Future-Proofing SOA Investments

An SOA integration platform that rapidly abstracts data and business logic into a consistent, efficient, and persisted canonical form, and then automatically exposes the resulting abstracted

business integration journal takeaways

BUSINESS

- Reduction of risk is key to faster adoption of SOA, Web Services integration efforts, and the realization of competitive advantage.
- Relying on Web Services standards alone for risk reduction and investment protection in SOA integration efforts may not be the best strategy.
- Shifting the burden of standards and technology support off of IT and onto the software vendor is a sound strategy for reducing risk and long-term investment protection.

TECHNOLOGY

- Higher-level abstraction at the center of SOA implementations reduces risk and complexity, increases flexibility in support of business processes, and speeds time-to-deployment.
- A universal services architecture approach to integrating services in an SOA provides a safety net for IT innovation in support of the business.
- Developers working within a universal services view of the SOA can use services in whatever format is needed for the task at hand, rather than being locked in to any one approach.

services into whatever technology context is required (Web Services, XML, Java, .NET, SQL stored procedures, and so on) provides significant longevity for SOA investments. This “universal services architecture” approach eliminates the complexity associated with creating services from heterogeneous systems, both new and legacy, while avoiding the verbosity of XML as the exclusive abstraction model. It also remains closer to the native form of the underlying systems for easier management, added flexibility, and greater efficiency.

An implicit, shared metadata repository of all abstracted services is a key byproduct of this persisted, canonical form approach. The metadata repository contains the abstracted services of all the external integration touchpoints and all the “internal” components of the solution such as adapter definitions and configuration parameters, request/reply message formats, routing rules and data transformation maps, orchestration and composition logic, and so on. As all these components share the single abstraction and projection model

described above, they’re all readily accessible from within the development environment of choice, be it .NET, Java, or XML/ Web Services.

Greater Freedom

Freedom to choose the suitable service invocation mechanism and appropriate development tools for different orchestration and composition projects mitigates risk. It also empowers IT to more rapidly realize the benefits associated with an SOA. Through use of a platform leveraging a universal services architecture, technology lock-in is avoided and investments are protected.

Not Monkeying Around

Unlike the Yale researcher’s monkeys deciding how to get grapes, IT decision-makers have much at stake. They walk a tight-rope between standards and innovation, between maintaining the status quo and taking action, often with their jobs on the line. So it’s natural to want to avoid risk and loss. The question is, at what cost to the business?

Real money is at stake here. Better process and information

integration between suppliers in the U.S. automobile industry supply chain, for example, could save \$5 billion a year, according to the U.S. National Institute of Standards and Technology. In the electronics industry, that figure is nearly \$4 billion per year. In healthcare, 95,000 people die needlessly each year in the U.S. due to medical errors, many of which could be eliminated through integrated information systems deployed at the point of care.

The universal services architecture approach discussed in this article can provide a safety net for IT stakeholders. With this technology in place, IT decision-makers can be bolder in supporting the business and driving business growth through IT innovation.

bij

About the Author



Trevor Matz is managing director, Application Integration, at InterSystems Corp., which develops and markets the Ensemble universal integration platform, and the Caché post-relational database.

e-Mail: Trevor_Matz@InterSystems.com

INTERSYSTEMS