

The Impact Of Service-Oriented Architectures On Data Access and Integration

BY SUSAN MESSEHEIMER AND CAROL WEISZMANN

Service-oriented architectures are changing the face of enterprise IT, provoking concern about how SOAs will affect database operations.

We contacted several developers and managers in industries ranging from health care to finance and asked them what they wanted to know from vendors about the impact of service-oriented architectures on data access and integration. From this we derived seven questions.

What do SOAs—which seem focused on the application level—mean for data access and integration?

A lot, according to these vendors. Indeed, data integration is “a key component” of SOAs, according to William Hardie, senior director of database product marketing at Oracle. “Different databases or data stores can exchange data with one another using open standards that include Web services. And having a consistent data model enables seamless information access.”

Key to SOA success is standardizing data representation and access, according to several vendors. “The way in which data is formatted and stored—case sensitivity in names, use of dashes in credit-card numbers, etc.—needs to be fairly consistent for successful SOA implementations,” noted Quest Software product architect Guy Harrison.

Microsoft’s SQL Server general program manager, Roger Wolter, believes that the isolation, flexibility and reusability benefits of SOAs “often require restructuring the underlying database.”

At Sleepycat Software, maker of the Berkeley DB open-source developer database, the view is that integration will be at the application or services level, not at the data level. “Each component needs to manage its own data reliably,” explained Sleepycat marketing vice president Rex Wang. “In a



sense, data silos are the norm in an SOA world.”

Some SOA models call for a data tier that is service-enabled, asserted Computer Associates senior architect Paul Lipton. The advantage: Data services are abstracted from the business logic. “If properly designed” he said, “this frees the application tier from having to intimately know the data sources—and probably the APIs of those data sources. There are also potential benefits in terms of simplicity and abstraction from data source specifics.”

Rod Smith, IBM’s vice president of software group emerging technologies, talked about an enterprise information bus that would accompany an enterprise services bus, providing “access to a virtualized world of information and a

means to manipulate that information.”

Are we finally getting away from the one-giant-database mantra in favor of cheaper, faster interoperability fueled by Web services?

Microsoft’s Wolter thinks so. “Building loosely coupled services often requires breaking the tight integration of data so the services can be distributed across a number of independent databases,” he said. But others dispute this view: CA, IBM and Oracle see a world with both monolithic and distributed databases, depending on the needs of the enterprise.

J. T. Taylor, senior director of XML business integration at Software AG, noted that few organizations can “afford to rip and replace, which is the first prerequisite for a single ‘monster’ database.”

Quest’s Harrison sees utility/grid computing models offering the promise of being able to deal with “what appears to be one giant database, even if the actual data is spread across multiple physical instances.”

Will the implied virtualization of some architecture layers mean dynamic allocation of resources, and hence improved efficiency of the computing environment?

Yes, yes, yes, agreed CA, IBM, Oracle and others. Still others cited conditions: “Only if you build the infrastructure to make use of it, and make sure that the ability to dynamically allocate is supported throughout your stack,” said Zack Urlocker, open-source database devel-

► continued on page 32

The Impact of SOAs on Data Access, Integration

< continued from page 31

oper MySQL's marketing vice president. "I think we are moving to a more on-demand, needs-focused data center," he said. "Computing power should be like electricity; the same is especially true for your database needs."

Software AG's Taylor thinks this type of capability is some distance in the future. "In order to make this work, other factors must first be addressed in a practical way—namely, security and accountability," he said. "Dynamic allocation also requires some smart piece of software, similar to a grid operating system, that simply doesn't yet exist."

Quest's Harrison concurred. "Mission-critical systems probably cannot take the risk of relying on a grid yet. Most computing grids are homogenous, while the true benefits will only be realized when a truly heterogeneous grid emerges," he said.

"We think virtualization at the storage and I/O layers is practical now, as is virtualization of some elements of the presentation layers," he continued, talking of dynamic provisioning of Web servers, for instance. "However, in the middle tier, only limited opportunity for provisioning RDBMS and app servers exists today, and those opportunities only exist when they are all of the same type."

What are the implications of putting Web services engines directly into the database (which major database vendors are doing)? How does this impact n-tier architectures?

Not surprisingly, the major database vendors take a different view of this issue than their smaller competitors. Sleepycat's Wang described it as "simply their attempt to differentiate and avoid commoditization."

The spokesmen from Quest and Microsoft take Wang's side. "We don't believe any large-scale take-up of Web services direct from the database will occur—at least not in a business-to-business or business-to-consumer context," said Harrison. "Security and authentication issues are tough enough as it is."

Microsoft's Wolter points to situations where handling Web services on the middle

tier allows for better overall application scalability, leaving the database server to focus resources on the database itself. "Also," he added, "if the Web services being used are callable directly over the Internet, you generally want the database behind a couple of firewall layers, so you need to have the Web services layer operating in the DMZ. I see the in-



built Web services stacks used primarily as a way of exposing the database to internal applications written on platforms that can't easily get to the database directly."

CA's Lipton said the embedding of Web services in the database introduces a greater use of database services not only for data access but also for application services. But, he said, "from a Web services and protocol perspective, this will not be as common as you would suppose."

Why? Even though the protocols and XML tags are used in Web services information, semantic issues remain because applications often treat the same data differently.

"From a database perspective, as more companies move to J2EE and .NET-developed applications, custom development by database type lessens," said Lipton. "No longer do developers code in database-specific SQL like T/SQL or PL/SQL; instead they are now invoking ANSI SQL data requests through JDBC connections, application servers or with XML. Database vendors are responding to this programming paradigm shift by not



Some SOA models call for a data tier that is service-enabled. 'This frees the application tier from having to intimately know the data sources.'

—Paul Lipton, senior architect for Computer Associates

only supporting Web services, but also embedding them within the database to improve performance."

Will database vendors be changing their pricing models (which some feel pose a major barrier to innovation)?

Pricing models are changing already. "The general trend for about five years," said

Quest's Harrison, "has been to lower overall cost of database ownership because of commoditization of base RDBMS capabilities, rough parity between the big three database vendors, and emergence of open-source alternatives."

Oracle, meanwhile, urges customers



'We are moving to a more on-demand, needs-focused data center. Computer power should be like electricity; the same is...true for your database needs.'

—Zack Urlocker, marketing vice president for MySQL

to look beyond list price. "Always look at total cost of ownership to determine their return on investment," advised Oracle's Hardie.

How do you see the major vendors responding to solutions offered by new pure-play entrants?

Here, database vendors are in consensus—up to a point. Software AG and Quest spokesmen both indicated that larger vendors generally do not feel competitive threats from smaller competitors, which lack the stability and credibility needed to attract large accounts.

But as the database industry becomes more commoditized, believes MySQL's Urlocker, "you will see the large vendors focus on other areas, such as the application stack or services. Even Oracle, which is No. 1 in the database market, has placed a key part of their bet on the future of applications like PeopleSoft."

Oracle explained its focus as larger than just systems connectivity using Web services. "Oracle is delivering a complete SOA life-cycle management platform built on open standards with support for XML, BPEL, SOAP, WSDL, UDDI, J2EE, JCA and

JMS," said Ashish Mohindroo, product director for Oracle database product marketing.

What are the main barriers to successful execution?

The vendors we contacted see plenty:

- "Islands of management" resulting in lack of an integrated view of the infrastructure.
- An information management philosophy that doesn't fully acknowledge the heterogeneity of information assets.
- Inadequate business justification for the project.
- Disparate or inconsistent representations of data.
- Poor design.
- Inappropriate application choices.
- Attempting to do too much at once.
- Deployment problems, often because of lack of familiarity with new technology.

And what can be done to overcome these barriers? Microsoft's Wolter counsels choosing your SOA projects very carefully. "There are classes of applications that will benefit greatly from SOA concepts and other applications that don't fit the SOA model well," he said. "The secret to success lies in

closely analyzing the application to determine whether the potential benefits of using SOA justify the implementation costs."

Software AG's Taylor thinks SOAs should be implemented with tactical, business-driven projects. "Fortunately, a particular instance of SOA can be implemented incrementally and even in parallel to other SOA projects," he noted.

Oracle's Hardie observed that an integration project requires more than technology to be successful. "The project needs to be clearly scoped and should have organizational buy-in from all departments that own the systems that are going to be integrated," he said. "From the technology perspective, companies need to base their integration solution on open standards rather than proprietary technologies. In terms of SOA, they need to consider the complete life-cycle management of Web services and not just focus on connectivity and orchestration."

MySQL's Urlocker offered some implementation suggestions: "Get accustomed to new technology by using it for new projects first," he advised. "Go back and replace existing infrastructure only after you feel comfortable with newer technology and understand how to deploy it. Doing too much at once does sometimes work, but the upheaval costs are high." ■