Illuminating the Dark Side of Web Services[1]

Michael L. Brodie

Chief Scientist
Verizon Communications
USA
Michael.Brodie@verizon.com

1. Panelists[2]

- Frank Leyman, IBM Distinguished Engineer, IBM Software Group, Germany
- Christoph Bussler, Digital Research Institute, Ireland

2. The Next Generation of Computing

Web Services are widely heralded as a step to the next generation of computing and a basis for resolving integration, one of the largest IT challenges. With essentially all vendors supporting Web Services and considerable focus on Web Services, it may appear as if Web Services are maturing consistent with analyst projections for the 2003 to 2005 period. The reality is quite different. Web Services are in their infancy. Designing, developing, and deploying a Service-Oriented computing model over the Internet is a massive undertaking. Having understood the potential of Web Services, like seeing the Moon on a clear night, it is now time to illuminate the dark side of Web Services. The purpose of this panel is to review the status of the development and usage of Web Services and identify significant technical challenges to which the database community should contribute.

Web Services has much to learn from the development of database management systems (DBMSs) and the DBMS community has much to contribute to realizing Web Services. Ted Codd's classic 1970 paper defined a complete computational model, the relational data model including the relational calculus. A decade of work by the major software vendors and the emerging data management research community was required to develop and implement the infrastructure, languages, and techniques required to produce the first commercial Relational DBMSs (RDBMS). It took another decade for RDBMSs to become robust and reliable enough to support large-scale industrial applications. Hence, it took two decades of research and development for RDBMS technology to go from concept to realization in robust, scalable technology. Will it take two decades to realize the Web Services vision?

Web Services has much to learn from the failures of previous Service-Oriented Architectures (SOAs). In the 1980s, several distributed computing proposals emerged including the Open Software Foundation's Distributed Computing Environment (DCE), the Object Management Group's (OMG's) Common Object Request Broker Architecture (CORBA), Microsoft's Distributed Component Object Model (DCOM), as well as several distributed DBMS prototypes and products. These distributed computing proposals where part of the widely accepted notion of an SOA based on modularization, encapsulation, and re-use in which services could be invoked remotely and transparently across a distributed computing environment.

OMG's CORBA, one of the most successful SOA proposals, was defined in 1989 in terms of the OMA Vision and Architecture. The first CORBA specification, published in 1991, was relatively complete conceptually. It included the CORBA Object Model, the Interface Definition Language (IDL), and a core set of application programming interfaces (APIs) for dynamic request management and invocation (DII) and an Interface Repository. Although the first CORBA compliant products emerged in the early 1990s, it took almost a decade for ORBs to become robust and reliable for large-scale applications. These products implemented only a small number of the many OMG-specified component services.

CORBA failed to gain support from key vendors, hence failed to obtain development resources and was subsequently not widely adopted, despite successful applications, e.g., in telecommunications. Consequently, it does not leave a large legacy of re-usable, robust SOA-enabling development tools, languages, and infrastructure, let alone libraries of re-usable solutions or services. The emergence of competing distributed computing / SOA

Proceedings of the 29th VLDB Conference,
Berlin, Germany, 2003

proposals, such as Web Services, may lead to the end of CORBA compliant ORBs as a technology.

Many SOA concepts were experimented with in the OMG community. One closely related to the Web Services vision was the OMG Trader. The Trader was intended to permit objects (services) to be published or declared to one or more traders so that systems or objects seeking such services could dynamically and automatically (without human intervention) find and invoke those services. Had the Trader worked as planned, Web Services, as currently envisaged, would be far more tractable. As it is, the same hard problems remain.

Web Services has miraculously gained the acceptance of all of the major software vendors, the analyst community, and the IT consumers. Will almost complete support be enough to overcome the failures of past SOAs?

2 Web Services Emergence

Based on the emergence of the web and of XML as the standard application-to-application protocol for data description, in 1998, and on the potential of electronic marketplaces, five major vendors (e.g., IBM, Microsoft, Oracle, HP, and Sun) agreed to support Web Services standards: XML, Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), and Universal Description, Discovery and Integration (UDDI).

In late 2000, based on the agreements led by Microsoft and IBM, Web Services were introduced as a standards-based remote service invocation for the Internet. By mid 2001 middleware and other products from these and other vendors supported the standards. Hence, Web Services went from concept to support in products of the above standards within one year.

However, the Web Services vision is much bigger than standards-based remote invocation. It includes the well-understood and highly desirable objectives of the 1980s SOA proposals. Web Services are intended to provide standards-based mechanisms for static and dynamic discovery, composition, and invocation of simple and composite services within an enterprise, between partners, or with enterprises and customers, Internet-wide. There will be scaleable and dynamic (i.e., automatic) means of registering and maintaining service descriptions in public and private directories; discovering services that meet requirements (a la Trader); and composing complex services from simpler services. Hence, the computing world will consist of service providers and service consumers. Providers can develop new services or expose legacy services and make them available within an enterprise, to partners, and to customers Internet-wide.

3 Web Services Benefits

Based on the simple standards and on the Internet, the technical benefits of Web Services include flexibility, low cost, and universal access Internet-wide. The technical benefits, if realized, clearly place service-oriented computing and SOAs within reach. The business benefits to could be profound. The re-use of existing services currently embedded in legacy applications could facilitate the simplification of existing infrastructure, through service standardization and elimination of redundancy; the modernization of infrastructure and applications by replacement or enhancement of services; and, of course, the biggest application challenge of them all, integration. Thus Web Services, if realized, will enable building and controlling the value of existing information systems assets through radical simplification, modernization, and enhancement of computing infrastructures and applications.

To derive more value from these information assets and due to the simplicity, flexibility, and low cost of Web Services, one might expect enterprises to rapidly automate and enhance services within an enterprise, as well as those provided to customers and partners, to achieve more flexible management of customer-vendor networks (e.g., adding and deleting vendors from supply chains). The benefits are so compelling that one might overlook the challenges in realizing the vision.

4 Web Services Predictions

Due to the universal appeal of the potential benefits of SOAs and to the unique industry-wide support of Web Services, analysts, such as Forrester, Gartner, and Yankee made ambitious projections for Web Services development and adoption.

Gartner predicted in January 2001 that Web Services are "projected as being in mainstream usage within enterprises by 2002 and between enterprises by 2003/2004." In 2002, The Yankee Group predicted that in 2001 legacy connectors would be re-engineered leading to Web Services; in 2002 XML-RPC would be adopted behind the firewall; in 2002/3 there would be inter-enterprise usage between existing partners; in 2003 there would be dynamic discovery and integration within the enterprise; and finally by 2005 there would be dynamic discovery and integration with outside partners. In March 2002 Forrester predicted, following read-only Web Services in 2002, trusted transactions in 2003; and coordination between partners in 2004. In early 2002 Gartner predicted that "By 2005, 50% of all transaction delivery network and electronic data interchange value-added networks will implement Web Services and thus will become Web Services networks (0.7 probability)." In November 2002 Forrester predicted that "Web Services foreshadow the next big thing for the Internet: a network of things, not people. By 2006, the bulk of the Internet's traffic will be digital chatter between software, not e-mail or Web pages aimed at human users."

The strength of vendor promises for Web Services supported by analyst schedules for their development and role out, plus the widespread discussion of Web Services in the trade press and research community led to the false impression that Web Services are well on their way to maturity.

5 Web Services Products

Whereas most middleware and related products support the four Web Services standards, much of the related technology is in research and development. Hence, there are no end-to-end product suites to support the life cycle of Web Services design, development, deployment, management, and evolution. The first Web Services-based offerings from major vendors are powerful integration platforms with little Web Services support than the four standards. Many small vendors are developing innovative solutions for the missing Web Services components. These will necessarily be proprietary, contrary to the standards-based solutions required for an Internet-based SOA. Hence, current products provide little more than support for the basic standards.

Although Web Services are standards-based and despite claims for Web Services as a universal interoperability standard, not all Web Services products are interoperable, e.g., .NET and J2EE based products. Hence, an interoperability standard, WS-I (Web Services Interoperation), is being developed.

6 Web Services Usage

The usage and deployment of Web Services is very different from analyst predictions, almost none of which have materialized.

Although many enterprises are interested in Web Services (65% in an April 2003 Gartner survey), most are early in investigation phases. The survey found that participants used Web Services standards as follows: 87% of respondent use XML; 31% SOAP; 3% WSDL; and 14% use UDDI. Most Web Services trials are small and restricted to fine grained, read-only (e.g., get Customer Service Record), services in a single application domain (e.g., telecom Billing) with trusted applications within an organization. Although a few sophisticated, experimental Web Services developments have been reported, e.g., by BT Impact, recent surveys report that there is almost no inter-enterprise usage of Web Services, dynamic discovery, or composition. Directory entry and maintenance is manual with hard coded invocations versus dynamically discovered services. More importantly, security solutions are hand coded. BT Impact, which reported success in developing fixed (e.g., point to point) Web Services within BT, experienced such difficulty that they launched a business to assist others in Web Service development and deployment.

Verizon Communications' 3 year old Web Services program has identified 100s of the 1,000s of potential services to expose as common services within Verizon. Deployment is underway, some via Web Services using an internally developed Web Services management system, ITWORKBENCH. As of mid 2003 two major applications in different business units are in production using ten Web Services and all four standards. Coordination between external partners is unlikely for some time. Verizon's first objective is infrastructure and application simplification through common services. In a more ambitious project Verizon has deployed an advanced set of communications services to its customers in a product called Digital Companion that is based on Microsoft's .NET and on the Sun ONE middleware platform.

In summary, Web Services usage and deployment is in its infancy. The few enterprises that have deployed Web Services have done so in the simplest form, RPC over a LAN between trusted partners with hand coded registration, invocation, and management.

7 Technology Challenges

Although the success of Web Services is in part based on simple standards, the simple standards provide very little of what is needed for a scalable, safe, reliable solution for the execution of Web Services over large networks. At the high level, there is no computation model such as offered by Codd's relational data model and calculus. There is no service-oriented architecture, comparable to a DBMS architecture. At a more detailed level, Web Services lack solutions for the basic components of the vision, automated registration and dynamic discovery and composition.

Current Web Service deployments immediately encounter the lack of infrastructure and management support that hinders even small-scale deployments. Early adopters first develop hand-coded solutions that quickly fail to scale for even small deployments. Some of these capabilities are being considered by one of the many standards efforts (named in parentheses), some are offered as proprietary solutions in emerging products, since standards do not exist. Proprietary solutions in a standards-based SOA are unacceptable.

Required infrastructure capabilities include: asynchrony, scalability, process management (BPEL4WS, BPML, WSCL, ebXML, etc.), mediation, reliable communications (WS-C), transactions (WS-T), deployment / provisioning, addressing (WS-Addressing), and interoperability WS-I (Web Services Interoperation).

Required management capabilities include management of auditing or accounting, faults, connections, configurations, Quality of Service (QoS) including availability, reliability, scalability, and robustness, security (SSL, SAML, XKMS,), and versions.

As with DBMSs it will take some time to understand the infrastructure and management capabilities required, including their relationship and the architecting of the managers and services into a Web Services (SOA) architecture. There are proposals to develop as components in a Web Services architecture specialized, highly optimized servers or managers for SOAP, XML, Directories / Registries, message management, security management, and Web Services Brokers. Web Service Brokers, like OMG Traders, receive service requests, identify the desired service and relay the service request to the intended service, possibly through other brokers. Brokers could then become the means of communication between Web Services, thus a focal point for many of the management and infrastructure capabilities.

8 Deeper Challenges

As daunting as the technical challenges may be, there are deeper, harder to resolve problems that must be addressed to achieve the benefits claimed for Web Services. Let's assume that the technical aspects of the Web Services vision succeed – an Internet-wide SOA infrastructure with the tools to support the complete Web Services Life Cycle. The deeper challenge is that of designing new (or decomposing legacy) applications in terms of re-usable base services that can be composed efficiently into higher level services to achieve any desired application or requirement that might be made of the base services (e.g., telecommunication billing, financial management, air traffic control). This should be done so that individual base or composed services can be modified, maintained, and enhanced as required. Even with the maturity of DBMS architectures there are debates as to where services (e.g., queues) should go. At the most basic level, how do you design services for re-use? What concepts, techniques, and tools are required to support the life cycle of web service-based applications?

There are deeper problems - semantic problems. How do you describe services so that they can be automatically identified, without human involvement, to achieve dynamic discovery? How do you design services so that (automated) consumers can understand what the service does and the associated business and other commitments required by or undertaken by the service? How do you determine if two services are equivalent for purposes of selecting between or eliminating redundant services? How do you automatically integrate services (applications) based on service descriptions? Currently, the above problems are solved by hand. Due to the number of Web Services even in simple applications, manual solutions are infeasible. For Web Services to fulfil the vision, service descriptions will have to permit a considerable degree of automatic discovery and composition.

These problems require semantic challenges to be addressed to an extent that has not been achieved in the history of computer science. Ontologies are the leading candidate for addressing some of the semantic challenges. Early attempts to develop industry or domain specific vocabularies and ontologies have failed for a number of reasons. The communities were unable to reach agreement on the ontologies. They were unable to accommodate innovation and required changes. And industries always work across domains or industries.

In recent years, ontologies have become trendy and widely referenced in the database and other communities, without a deep understanding of this difficult, complex area. The need for resolutions to some semantic problems is critical for the success of Web Services. The database community should appeal to the ontology community to avoid inadequate applications of ontological concepts and technology, e.g., treating ontologies as schemas.

9 Database Community Contributions

Starting in 2000 with less of a conceptual basis than the relational model had in 1970 and with a much grander vision and greater challenges, why is developing a new major computing infrastructure different this time? Will the Web Services vision succeed and if so when will they be ready for large-scale industrial use? Like the relational model, Web Services will require an entire computing infrastructure and extensive management support plus concepts, tools, and techniques to support the Web Services development life cycle. Unlike the relational model, Web Services has a grander vision including dynamic discovery and composition, scalability across the Internet and a vast number of services, interoperability across all platforms, integration of all services (data, applications, and processes), performance aspects involving the network and all involved components, and a producer-consumer accounting model, which will involve the QoS and business aspects never required for DBMSs.

What is different this time is the unprecedented cooperation and commitment of the major vendors and indeed of most vendors. This is leading to a commitment of resources required to address such an undertaking, a commitment missing in all previous SOA attempts. The commitment is to a standards-based SOA solution, but much needs to be done to achieve the vision. The database community is well positioned to contribute to the Web Services development.

With three decades of developing one of the worlds most reliable and scalable computing infrastructures, the database community has much to contribute to illuminate the dark side of Web Services. The database community should contribute directly in the areas of infrastructure, management, architecture, data management, transactions, modelling, integration, and of course scalability. Through years of experience in modelling and integration, the database community has a deep understanding of the significance of the semantics challenges in database and application integration that only hints at the challenge in the much grander Web Services vision. As with surgery, it may be best not to try this at home without significant professional advice from the ontological community.

---

[1] Title suggested by John Mylopoulos.

[2] Participants confirmed at publication time.