Gartner

Research

Publication Date: 3 January 2008 **ID Number: G00154463**

Service-Oriented Architecture Overview and Guide to SOA Research

Charles Abrams, Roy W. Schulte

This research provides an overview of service-oriented architecture (SOA) and a summary of how it is being adopted by real-world companies. It helps IT managers and business analysts become familiar with the concept as it is applied, and it provides references to other Gartner research that drills down into specific SOA-related topics in more detail.

Key Findings

- The use of SOA is accelerating in response to the growing need to more frequently change business processes, products and services. SOA was used, to some extent, in more than 50% of large, new applications and business processes designed in 2007.
- The majority of SOA projects succeed, and most companies that have started using SOA have continued with it. However, some companies have been disappointed by the low level of service-sharing ("reuse") that they have achieved, and some SOA projects have failed for a variety of reasons.
- Companies using SOA face challenges in governance, testing, configuration, version control, metadata management, service-level monitoring, security and interoperability. One of the greatest challenges is managing application logic and data in SOA service components that are spread out over multiple business units.

Recommendations

- Use SOA when designing large, new business applications, particularly those whose life spans are expected to be more than three years and those that will undergo continuous refinement, maintenance or enlargement. Use SOA to refactor and retrofit selected parts of legacy applications that can be used in new ways.
- Implement an SOA center of excellence or integration competency center in the IT department to coordinate the SOA and integration activities of diverse business units in the company.
- The decision to adopt SOA as a companywide architectural principle should be escalated to top management, rather than being confined within the IT department, because of its effects on the business organization and decision-making processes, as well as its significant long-term costs and benefits.

TABLE OF CONTENTS

ANALYSIS

1.0 What You Need to Know

SOA is a durable change in application architecture, like the relational data model and the graphical user interface. Application development tools from all the major software vendors enable and encourage the use of SOA. Architects and software engineers in user companies have a growing understanding of how to implement SOA. Most of the major packaged-application vendors have adopted SOA and are changing their products to implement SOA more broadly. Many IT and business managers have also come to demand SOA, further accelerating the transition.

Systems that are built to change are more valuable than systems that are built to last. In reality, systems that are built to change are the only ones that last. SOA is used to build systems that are intended to change. SOA application portfolios start small and are incrementally maintained, modified and expanded over time as business requirements change and additional business functions are brought into the scope of the service community.

However, changes of this magnitude in the fundamental structure of business applications do not happen quickly or easily. Mainstream user companies are gradually ramping up their use of SOA and are experiencing varying degrees of difficulty. These difficulties slow, but will not prevent, the eventual spread of SOA throughout the application portfolios of most companies. SOA was used, to some extent, in more than 50% of large, new applications and business processes designed in 2007. By 2010, we expect that more than 80% of large, new systems will use SOA for at least some aspect of their design.

2.0 The Service Concept

Companies implemented the service concept at the business level long before they used it in software. Many business functions in a modern enterprise are organized in the form of services — for example, companies use cleaning services, copy services, package and mail courier services, legal services, security services, personnel recruiters, auditors, advertising agencies, travel agencies and many other services (see "Five Principles of SOA in Business and IT" for a more thorough explanation of the service concept in business terms).

The same characteristics that make the service concept helpful in organizing business units are used in the design of SOA applications. However, using services in software development requires a degree of rigor and precision not always required in business-level services.

An SOA application adheres to the following five principles:

1. The system must be modular. This provides the obvious benefit of being able to divide and conquer — solve a complex problem by assembling a set of small, simple components that work together.
2. The modules must be distributable, able to run on disparate computers and communicate with each other by sending messages over a network at runtime.
3. Module interfaces must be clearly defined and documented. Software developers write or generate interface metadata that specifies an explicit contract so that another developer can find and use the service (this helps enable loose coupling).
4. A module that implements a service can be swapped out for another module that offers the same service and interface, because the interface designed is separate from the

module. This is an aspect of loose coupling and it enables incremental maintenance and enhancements.

5. Service provider modules must be shareable — that is, designed and deployed in a manner that enables them to be invoked successively by disparate service-consumer modules engaged in somewhat diverse, although partially related, business activities.

Any business application that implements these five principles is an SOA application. However, it is not safe to assume that all applications labeled "SOA" have actually met this acid test, because the IT industry uses many other definitions of SOA. Many definitions of SOA are not specific about what SOA is, so be cautious when engaging in SOA discussions.

In some contexts, SOA only means any modern, distributed application system. Sometimes, SOA is used as an umbrella term to refer to a vast set of concepts that were popularized around 2000. These other concepts may include business process management, the Web, event processing, policy management, metadata management, governance and others. However, these are separate dimensions that are not strictly linked to the five characteristics of SOA described here. These other concepts can apply to non-SOA applications as well as to SOA applications (although in different ways). Gartner believes that it is a mistake to gather all these concepts under the SOA banner. Each has its own set of principles, technologies, benefits, limitations and best practices that differ from those strictly associated with SOA. Admittedly, they are complementary and mutually reinforcing, and they happen to be emerging in the industry at the same time. Nevertheless, each of these should be understood in its own right and then implemented in combination with SOA and the other concepts, according to the needs of the business.

Gartner has published extensively on many aspects of SOA. We find it useful to organize our analysis into five phases of the application activity cycle, which reflect the work in which application leaders and professionals engage (see "The Application Management Activity Cycle"). The five phases are: strategize, evaluate, execute, review and innovate.

2.1 Strategize: SOA Adoption Considerations

The use of SOA is accelerating in response to escalating business requirements. Competition, globalization and technology advances are driving companies to change their products, business processes and prices more frequently than they did before the 1990s. The main benefit of SOA is that it reduces the effort and time needed to change application systems to support changes in the business (see "The Business Impact of Service-Oriented Architecture" and "User Survey Analysis: SOA, Web Services and Web 2.0, North America, 2H06").

Overall, market hype in favor of SOA is still high, but it is being countered by some developer and management disillusionment (see "Hype Cycle for Application Development, 2007" and "Hype Cycle for Application Infrastructure Middleware, Platforms and Architecture, 2007"). Some companies have been disappointed by the low level of service sharing ("reuse") that they have achieved. Some SOA projects have failed for a variety of reasons, including unclear objectives, lack of coordination between business units implementing SOA in isolation from each other and spending too much time modeling interfaces without delivering working systems. Nevertheless, the majority of SOA projects succeed, and most companies that have started using SOA continue with it and expand their use of it.

The changing perceptions of SOA have come about, at least in part, because more companies are actually using SOA in production. They are realizing first hand that it is not easy to do well. SOA applications are composed of many "moving parts," so they are inherently more complex than monolithic applications. SOA is not the cause of these problems, but moving to SOA causes these problems to surface. Compared with traditional monolithic or client/server applications, SOA applications are more likely to be:

- Spread across multiple computers in far-flung locations
- Composed of parts that are developed and managed by disparate, semiautonomous IT groups, often controlled by disparate business units
- Running on a mix of heterogeneous application servers, programming languages and operating systems
- Subject to frequent change because of volatile business requirements

SOA is part of the solution to these problems because it clarifies system design, isolates the modules from each other and increases the interface documentation. Parts can be changed incrementally without causing unintended side effects.

The IT department can execute small SOA proof-of-concept projects to validate the concepts and technologies without affecting the company as a whole, and at relatively low cost. However, the decision to adopt SOA as a companywide architectural principle has significant long-term costs and benefits, and it often affects the relationships between business units. This decision should be escalated to top management, rather than being confined within IT management (see "Building a Service-Oriented Architecture Business Case: Effective Communication Is the First Step" and "Applied SOA: Transforming Fundamental Principles Into Best Practices"). SOA changes the concept of what an application is (see "Defining 'Application' for Application Portfolio Management") and drives adjustments in how application portfolios are managed (see "Application Portfolio Triage: TIME for APM").

2.2 Evaluate: Planning and Designing SOA Systems

Gartner coined the term "SOA" and published the first reports in the industry on SOA in 1996 (see "'Service Oriented' Architectures, Part 1" and "'Service Oriented' Architectures, Part 2"). Companies that adopt SOA as a major part of their application architecture progress through four stages. Choosing a starting point for the SOA journey is far from trivial. The best approach is to proceed with a succession of small projects, linking each of them to business and technology metrics, and measuring project success against these metrics in a way that IT and business sponsors understand (see "SOA: Where Do I Start?"). A portal is often a logical and appropriate first step toward SOA.

SOA-based systems are assembled using components from packaged applications (called service-oriented business applications [SOBAs]), custom-written applications or externally hosted services (sometimes called "cloud" computing because the services can be located anywhere in the network "cloud" and accessed as software as a service). In many cases, SOA applications are a combination of two or all three of these. When choosing between packaged applications, companies should rate products that implement SOA more highly than those that do not use an SOA style.

Major software ecosystem giants such as IBM, Microsoft, Oracle and SAP; other technology vendors; external service providers; and end users are developing SOBAs (see "Key Issues for SOBA Strategy and Governance," "Criteria for Evaluating a Business Application Vendor's SOA Strategy" and "Evaluating IBM, Microsoft, Oracle and SAP Business Application SOA Strategies").

Pre-SOA applications may be retrofit with SOA interfaces using software "wrappers" so that they can be used as part of a new SOA application (see "Key Issues for Platform and Application Modernization, 2007"). However, do not discard non-SOA applications in favor of SOA applications unless there are compelling business reasons why the non-SOA application has become unsatisfactory. Continue to use non-SOA architecture styles for some new, tactical applications of limited size and complexity, and for minor changes to installed non-SOA applications.

SOA developers use industry standards wherever practical so that their applications can be composed of components that run on diverse platforms from multiple vendors. Web services are not required by the definition of SOA, but most SOA applications use at least some of the Web and Web services standards, particularly SOAP, Web Services Description Language (WSDL) and related standards. Gartner's "Hype Cycle for Web and User Interaction Technologies, 2007" shows the evolution and maturity of some of the standards and specifications that affect connectivity and Web services. Service component architecture (SCA) is an emerging set of standards that have the potential to significantly enhance the development and deployment of SOA applications. SCA provides a graphically based, metadata-centric approach to specify the design of SOA solutions using Web services policy.

Many of the new SOAP-based Web services (WS-*) specifications are complex, which seems to run counter to the simplicity originally envisioned for Web services. Companies must consider the role of simpler Web technologies in their SOA applications (see "Advanced Web Services Lead to the Next Generation of Enterprise-Class Computing"). Representational State Transfer (REST), which leverages the fundamental Web principles and technologies, such as URIs, XML and HTTP, remains the dominant approach to implementing SOA on the public Web. It is also used as an alternative to SOAP for some types of SOA processing, even in the virtual enterprise (see "Understanding and Applying the Design Differences Between WS-* Based Architecture and Web-Oriented Architecture" and "Applying WS-* Based Web Services and WOA Standards to Enterprise Application-to-Application Interoperability Challenges").

Business flexibility and innovation are enhanced as companies establish a business process platform (BPP) specific to their needs to enable process creation, deployment and monitoring. A business service repository is an essential ingredient in a BPP. The emerging generation of business applications and enabling software architecture marks a shift in the boundary between application functionality and supporting infrastructure (see "BPP Changes Infrastructure and the Business Application Vendor Landscape").

BPM is a natural complement to SOA (see "SOA and BPM Are Better Together"). Architects and business analysts have come to recognize the benefits of explicitly capturing the business process using process modeling and process management tools. Architects also recognize that the flow of control in a business activity or a whole business process should be organized into four distinct levels (see "Flow Management in SOA: One Size Doesn't Fit All").

SOA applications need a middleware infrastructure that is different from the middleware that was used in traditional monolithic and client/server applications. Companies that implement large, mission-critical SOA applications generally implement some form of enterprise service bus (see "The Enterprise Service Bus: Communication Backbone for SOA").

2.3 Execute: Implementing and Managing SOA in the Real World

Companies using SOA face challenges in governance, testing, configuration, version control, metadata management, service-level monitoring, security and interoperability (see "The Top Four Challenges of Effective SOA Management"). Governance technologies encompass Web services management and security tools, as well as repositories and registries (see "Technical Approaches to and Considerations for SOA Governance" and "Criteria for Evaluating a Vendor's SOA Governance Strategy").

One of the most challenging aspects of SOA is that the data and logic needed to carry out a single business activity or process may be spread out over multiple places (see "Succeeding With Multiple SOA Service Domains and Disparate ESBs"). There is no simple solution to this challenge, but companies that recognize the issue and address it thoughtfully can overcome it. One of the most promising strategies is to implement a group in the IT organization to coordinate the SOA and integration activities of the various IT groups in a company (see "Findings: SOA COE = ICC+ Governance Processes").

2.4 Review: Improving and Refining the Use of SOA

Successful SOA strategies view application systems as organic in nature. As noted earlier, SOA application portfolios start small and are incrementally maintained, modified and expanded over time as business requirements change and additional business functions are brought into the scope of the service community. SOA is a journey, not a destination (see "Twelve Common SOA Mistakes and How to Avoid Them" and "Use Full Life Cycle Management to Reduce SOA Downtime").

SOA does not run independently of the rest of the IT application portfolio — SOA application systems coexist with non-SOA applications. Furthermore, non-SOA applications can be gradually retrofit to leverage the benefits of SOA (see "Batch Processes Can Take Advantage of SOA"). SOA must become integrated with the daily operation of many aspects of the IT department, including its quality-assurance activities (see "Quality Assurance Practices Will Drive the Reuse of Services in SOA Environments") and its financial planning and budgeting activities (see "Questions on the Impact of SOA on Software Pricing From Gartner's IT and Software Asset Management Conference").

2.5 Innovate

The most significant benefit of SOA is that it enables the business to be innovative in its product offerings, services and business practices. Business managers should be able to add or change an SOA-based business process more quickly than they could business processes that are supported by non-SOA systems, all else being equal. SOA makes it possible for a company to modify its business architecture more quickly to respond to changes in customer demands and the market at large.

A secondary benefit of SOA is that it helps facilitate much of the innovation that is occurring in the IT industry. The concept of SOA continues to evolve. The first wave of SOA assumed an interaction between the consumer and service provider modules, almost exclusively employing a request/reply communication pattern reminiscent of traditional remote procedure calls. As SOA developers encountered new types of business requirements, they realized the need for fully asynchronous, event-driven architecture (EDA) relationships for some situations. The concept of SOA was broadened to encompass EDA design patterns to complement the familiar request/reply patterns (see "Advanced SOA for Advanced Enterprise Projects").

SOA and EDA are paving the way for another innovation — the growing sophistication of context-aware applications. Such applications modify their behavior by tapping context data. They are tailored to a particular customer or other user and a particular setting. Business applications collected and used context data for many years, but the means for doing this on a systematic and shared basis have been enhanced by leveraging SOA (see "Context Delivery Architecture: Putting SOA in Context").

Finally, SOA is also paving the way for other IT trends that can leverage the modular and distributed nature of SOA applications. For example, virtual computing (see "Future Software Scenarios: Uncertainties of Service-Oriented Architecture and Business Virtualization") and model-driven software development (see "SOA Is a Catalyst for Model-Driven Development") would not proceed as quickly (or, perhaps, not proceed at all) without the beneficial effects of SOA.

RECOMMENDED READING

Activity 1: Strategize

- "Five Principles of SOA in Business and IT"
- "The Business Impact of Service-Oriented Architecture"
- "User Survey Analysis: SOA, Web Services and Web 2.0, North America, 2H06"
- "Hype Cycle for Application Development, 2007"
- "Hype Cycle for Application Infrastructure Middleware, Platforms and Architecture, 2007"
- "Building a Service-Oriented Architecture Business Case: Effective Communication Is the First Step"
- "Applied SOA: Transforming Fundamental Principles Into Best Practices"
- "Defining 'Application' for Application Portfolio Management"
- "Application Portfolio Triage: TIME for APM"

Activity 2: Evaluate

- "'Service Oriented' Architectures, Part 1"
- "'Service Oriented' Architectures, Part 2"
- "SOA: Where Do I Start?"
- "Key Issues for SOBA Strategy and Governance"
- "Criteria for Evaluating a Business Application Vendor's SOA Strategy"
- "Evaluating IBM, Microsoft, Oracle and SAP Business Application SOA Strategies"
- "Key Issues for Platform and Application Modernization, 2007"
- "Hype Cycle for Web and User Interaction Technologies, 2007"
- "Advanced Web Services Lead to the Next Generation of Enterprise-Class Computing"
- "Understanding and Applying the Design Differences Between WS-* Based Architecture and Web-Oriented Architecture"
- "Applying WS-* Based Web Services and WOA Standards to Enterprise Application-to-Application Interoperability Challenges"
- "BPP Changes Infrastructure and the Business Application Vendor Landscape"
- "SOA and BPM Are Better Together"
- "Flow Management in SOA: One Size Doesn't Fit All"
- "The Enterprise Service Bus: Communication Backbone for SOA"

Activity 3: Execute

- "The Top Four Challenges of Effective SOA Management"
- "Technical Approaches to and Considerations for SOA Governance"
- "Criteria for Evaluating a Vendor's SOA Governance Strategy"
- "Succeeding With Multiple SOA Service Domains and Disparate ESBs"
- "Findings: SOA COE = ICC+ Governance Processes"

Activity 4: Review

- "Twelve Common SOA Mistakes and How to Avoid Them"
- "Use Full Life Cycle Management to Reduce SOA Downtime"
- "Batch Processes Can Take Advantage of SOA"
- "Quality Assurance Practices Will Drive the Reuse of Services in SOA Environments"
- "Questions on the Impact of SOA on Software Pricing From Gartner's IT and Software Asset Management Conference"

Activity 5: Innovate

- "Context Delivery Architecture: Putting SOA in Context"
- "Future Software Scenarios: Uncertainties of Service-Oriented Architecture and Business Virtualization"
- "SOA Is a Catalyst for Model-Driven Development"
- "Advanced SOA for Advanced Enterprise Projects"

REGIONAL HEADQUARTERS

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters
Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters
Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters
Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters
Gartner do Brazil
Av. das Nações Unidas, 12551
9° andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509