Service Oriented Architecture: Tools and Technologies

ZAIGHAM MAHMOOD
School of Computing
University of Derby
Kedleston Road, Derby, DE22 1GB
UK

Abstract: - Service Oriented Architecture (SOA) is an emerging architectural style for developing and integrating enterprise applications. Businesses are required to be agile and flexible and IT managers are being asked to deliver improved functionality while leveraging existing IT investment. Globalisation, tighter economies, business process outsourcing and ever increasing regulatory environments are forcing the large enterprises to transform the way they provide their business and services. In this context, SOA promises better alignment of IT with business, seamless integration of business functions and reduced costs of development and the industry is keen to embrace this new architectural style. Realising the opportunity, major enterprise software vendors have released a number of development tools. However, a majority of these are difficult to use and do not deliver the benefits they claim. In this paper, we introduce the SOA approach, present the benefits it offers, discuss some of the frameworks, tools and products offered by major software vendors and provide general guidance for building and implementing SOA. The objective is to provide enough background information that enterprises wishing to embark on the road to SOA have a better understanding.

Key-Words: - Service oriented architecture, SOA, Enterprise applications integration, Web services, Service orientation, XML.

1. Introduction

Service Oriented Architecture (SOA) is an emerging architectural style for developing and integrating enterprise applications. It is an organisational and technical framework to enable an enterprise to deliver self-describing and platform independent business functionality [1] providing a way of sharing business functions in a widespread and flexible way. Knorr and Rist [2] define SOA as a broad, standalone and standards based framework in which services are built, deployed, managed and orchestrated in pursuit of an agile and resilient IT infrastructure. British Computer Society's definition suggests that *SOA is about the evolution of business processes, applications and services from today's legacy-ridden and silo-oriented systems to a world of federated businesses, accommodating rapid response to change, utilizing vast degrees of business automation* [3]. This architecture aims to provide enterprise business solutions that can extend or change *on demand* as well as provide a mechanism for interfacing existing legacy applications. It is, therefore, being seen as an approach to provide a closer alignment between a business and its IT systems.

In this paper, we first establish the need for SOA and mention the benefits offered by the SOA as well as the inherent issues. Then, we outline the SOA framework, discuss the available tools and technologies and present suggestions for building and implementing the SOA paradigm. The last section presents a summary and conclusions.

2. The Need for SOA

Enterprises have invested heavily in large-scale applications software such as ERP (enterprise resource planning), SCM (supply chain management), CRM (customer relationship management) and other such systems to run their businesses. The infrastructure is often heterogeneous across a number of platforms, operating systems and languages. There is often a huge duplication of functionality and services resulting in a waste of valuable resources and poor response times. Increasingly, the business and IT managers are being asked to deliver improved functionality of services while leveraging existing IT investment and provide continuous process improvement, new channels of business, business agility as well as business architecture that is *organic* in nature [4].

One solution is to develop architectures that allow easy integration of the existing and new enterprise applications. As Savvas [5] reports: *2007 will be characterised by an increasing migration to SOAs and the introduction of lean manufacturing principles*. In the US, 64% of Chief Information Officers plan to implement SOAs in 2007, which reflects a clear growth in the adoption of SOA in organisations [5].

SOA provides an opportunity to achieve broad-scale interoperability while offering flexibility to adapt to changing technologies and business requirements. If implemented correctly, SOA offers the following benefits [6, 7]:

- Loosely coupled coarse grained services
- Seamless connectivity of applications
- Location transparency
- Alignment of IT with business needs
- Enhanced reuse of assets/applications
- Parallel and independent development
- Better scalability, ease of maintenance and graceful evolutionary changes
- Reduced cost of development
- Reduced vendor lock-ins.

However, SOA also requires a large upfront investment in technology as well as a different mindset. There are also numerous issues, due to the very nature of service-orientation [8, 9, 10]. These issues can be summarised as follows:

- Coarse granularity: This may mean that testing and validating every combination of every condition in a complex service may well become humanly impossible. Also, a generic service, because of its coarse granularity, cannot be easily optimised for efficiency [9].
- Loose coupling: It is an architect's dream, however, as Fowler [10] puts it, it can become a *developer's nightmare*.
- Integration of services: This can be a complex task especially when there is a lack of skilled personnel.
- Service interoperability: This can become a serious issue when web services exchange SOAP messages over HTTP, encapsulating XML data in heterogeneous environments on distributed systems.
- Evolutionary development: If applications continually require additional functionality, and these requests are granted, the entire system may become unstable [9].

3. SOA Elements

In a SOA, the business and technical processes are implemented as services. Each service represents a certain functionality that maps explicitly to a step in a business process [11]. An important aspect is the separation of service *interface* (the WHAT) from its implementation or *content* (the HOW). The interface provides service identification, whereas, the content provides business logic.

Zimmermann [12] suggests three levels of abstractions within SOA:

- Operations: units of functions with specific interfaces
- Services: groupings of operations
- Business processes: actions/activities to perform business goals by invoking multiple services.

SOA uses a *publish-find-bind-execute* paradigm as shown in Figure 1. *Service Providers* build services and offer them over the network. They *register* services with service brokers and *publish* them in distributed registries. Each service has an interface, known as *contract*, and functionality, which is kept separate from the interface. The *service consumers* search for services and, when found, a dynamic *binding* is performed. The service provides the consumer with the *contract* details and an *endpoint* address. The consumer then *invokes* the service.

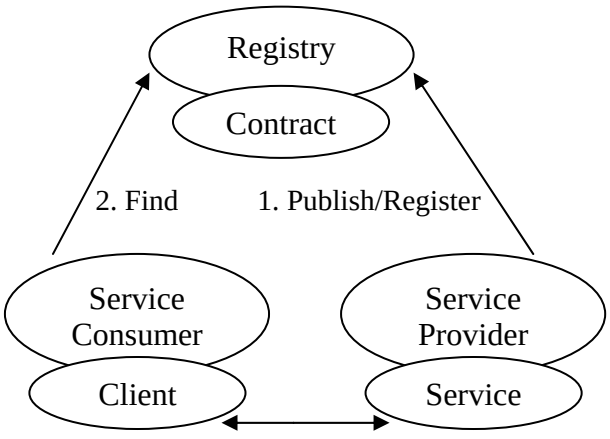

Figure 1: *Publish-Find-Bind-Execute* paradigm

4. SOA Tools and Technologies

In SOA, services are usually implemented as Web Services (WS) and delivered using technologies such as eXtensible Markup Language (XML), Web Services Description Language (WSDL), Simple

Object Access Protocol (SOAP) and Universal Description Discovery and Integration (UDDI).

XML is fundamental to web services. It provides a way to describe information. WSDL is used to describe WS interfaces, which define operations and then binds them to one or more protocols. SOAP defines an *envelope* and rules for representing information sent in that envelope. SOAP messages are commonly conveyed using HTTP. UDDI stores registrations describing WSs and provides unique names for elements in the registration.

Many proprietary SOA tools and frameworks have also been produced for the development of WSs and implementation of SOA. There are generally three distinct generations of these [13, 14]:

- First generation - Basic tools that retrofit the old to make them look like the new.
- Second generation - Simple tools that enable developers to build simple solutions using the new technology.
- Third generation - Efficient tools for building complex solutions.

Currently, the available tools fall into the first two categories. Majority of these are difficult to use and do not deliver the business benefits claimed. They lack vital capabilities like configuration control or testing prior to deployment. Hohpe [13, 14] believes the next generation tools will provide facilities for testing and debugging as well as provide support for monitoring and management.

Ward-Dutton [16] describes four main areas of functionality that customers need to look for when investigating tools for SOA initiatives.

- Service enablement - functionality to create service interfaces based on the capabilities of existing applications.
- Orchestration and composition - functionality to provide aggregate services and create *composite services*.
- Lifecycle management - functionality to support development and operations teams to provide high-quality consistent service experience to consumers.
- Service development - functionality to provide ability to design services from scratch or from existing applications and systems.

Here is a small list of recently released products provided by some of the major vendors.

4.1 BEA SOA 360

In late 2006, BEA Systems, a company specialising in the development of enterprise infrastructure software, announced, what they call, SOA 360 Degrees platform [18] to provide a unified SOA platform for business transformation and optimisation. It consists of three product families, as follows:

- Tuxedo - for service enablement of legacy assets
- WebLogic - for the development and deployment environment to create, expose and deploy services
- AquaLogic - for orchestration, management, discovery, security and visualization of business processes.

The SOA 360 is built on the company's microService Architecture (mSA) supported by a free-form collaborative environment known as BEA WorkSpace 360. The WorkSpace is intended to fundamentally change the way that both business and IT professionals collaborate and work individually across the extended enterprise. BEA believe that this product will deliver a truly shared workspace for the business analyst, enterprise architect, service/application developer and the IT operations professionals. The product provides comprehensive SOA lifecycle management functionality where participants can actually see how their efforts are aligned to impact the overall success of the business.

The mSA is based on the principles of SOA and the concept of a Service Network viz: separation of concerns and modularity. It is event-driven, using notification services to publish/discover components and other microServices. The principle of *separation of concerns and substitutability* helps to enable the mSA to evolve with the needs of the enterprise.

4.2 Eclipse SOA Tools Platform

Eclipse is an open source foundation backed by over 100 of the biggest names in enterprise software. It provides federated support for software development tools. It offers a SOA Tools Platform (STP) based on its Service Component Architecture (SCA) model [19]. The STP project aims to build frameworks and extensible tools that enable the design, assembly, deployment, monitoring, and management of software designed around the philosophy of service-orientation. The project is guided by the values of transparency, extensibility,

vendor neutrality, agile development, and standards-based innovation. Its sub-projects include [19]:

- STP Core Frameworks (CF): To define models conforming to the SCA specification for service assembly as well as Java language components for SCA syntax support.
- STP SOA System (SOAS): To provide tools/frameworks for assembling, packaging and deployment of services.
- STP Service Creation (SC): To handle the management of the relationship between the SOA model tooling (provided by STP) and the actual implementation tooling: goal is to support the development in an agile way.
- STP BPEL 2 Java (B2J): To provide tools to translate BPEL (Business Process Execution Language) into executable Java classes and define a standard framework for their deployment.
- STP BPMN (BPMN): To provide a set of tools to model business process diagrams using the Business Process Modelling Notation (BPMN) and allow validation and generation of BPEL artefacts from those diagrams.

4.3 IBM's WSDM for SOA management

In 2006, IBM released WSDM (Web Services Distributed Management) development kit [20] which seeks to unify management infrastructures by providing a vendor, platform, network, and protocol neutral framework for enabling management technologies to access and receive notifications of management-enabled resources. Built upon a standardized suite of XML, it includes the following:

- An Autonomic Manageability Endpoint Builder - to allow developers to build WSDM interfaces for resources such as servers, applications and printers.
- Endpoint Simulator - to create a WSDM based testing environment.

The Manageability Builder provides support to create XML definitions, on which the endpoint run time can operate in a WSDM-compatible manner. The manageability endpoint run time is a web services based run-time environment for WSDM and can run on any platform with a SOAP engine.

The WSDM toolkit is freely available from IBM's AlphaWorks site: www.alphaworks.ibm.com. For an introduction to WSDM, refer to [21].

4.4 CapeClear's SOA Tools

CapeClear Software [22] is a privately held firm with offices in the US and the UK. They have developed an ESB (Enterprise Service Bus) Platform that delivers on-demand integration reliability, scalability and performance to connect content, services and software across the Internet using web services. CapeClear ESB is built around open standards and designed to work with a company's existing infrastructure to provide implementation of SOA in a simple way. It supports core Web services protocols: SOAP, HTTP, XML, XML Schema, WSDL, UDDI. WS Interoperability (WS-I) is ensured through conformance to the WS-I Basic Profile, WS-I Attachments Profile and WS-I Basic Security Profile [23]. Additional features are provided through support for WS-Security, WS-Addressing, WS-RM and BPEL.

CapeClear Software also has a number of other SOA tools, including a SOA editor to provide an environment to develop web services and simplify the creation of WSDL [24]. This Java based tool includes wizards for development tasks, support for web services standards, testing tools, and documentation to help the development team. Other features include: 1) the ability to edit, validate and print WSDL files; 2) full support for XML Schema, WSD and SOAP and 3) the ability for developers to include MIME and non-XML attachments in service definitions. The Editor is freely available and runs on Linux, Windows and Solaris.

4.5 Stampede Web 2.0 Series

In 2006, Stampede Technologies, based in Ohio, US, announced the release of the Stampede Web 2.0 Performance Series [25]. The series provides functionality and business value for developing SOA applications and includes:

- Acceleration tools for SOA
- Asynchronous JavaScript/XML and XML applications featuring XML acceleration
- XML threat management and transparent client support for applications developed using AJAX.

The client services include XML document differencing, HTTP and TCP management, large document streaming via their TurboStreaming technology and SSL client-side termination. The product is offered in two configurations: 1) software acceleration for environments that require limited XML capability and 2) hardware acceleration for high-end XML implementations with intense

processing requirements. The hardware option features XML content inspection, XML well-formed-ness checking and message anomaly detection.

5. SOA Implementation

Implementation of SOA requires a shift in how we compose service-based applications while maximising existing IT investment [15]. It requires a shift from writing software to assembling and integrating services. Also, to support the goal of SOA, the infrastructure must support flexibility, heterogeneity and distributed development [6].

For a successful transition to a SOA, life cycle stages can be viewed as being the following:

- Development: of loosely coupled application components as services.
- Discovery: by organising a service directory, to act as 'yellow pages'.
- Integration: of services with applications and other services.
- Orchestration: of services to 'sequence' them to fulfil a particular business task.
- Deployment: of orchestrated services for the 24-hour-7-day availability.
- Monitoring: of processes in real time and analysis and resolution of issues.
- Management: of the entire process.

As for the environment, infrastructure and controls, there is a need for the establishment of the following key elements for successful transition to a SOA:

- Governance framework: consisting of rules and policies to ensure that risks are managed, duplication is avoided, processes are discoverable, standards are followed and system changes appropriately controlled.
- Process change mechanism: to align business processes to IT services in a more explicit manner so that information and data are created, updated and managed more efficiently – so that processes can adopt more easily to the changing environment.
- Business process management: to make services more visible in business processes and manage business processes in real time, avoiding errors and duplication – and to construct or *orchestrate* new processes from existing services.

In simple terms, the organisations can use the following strategy [14]:

- Start with the main business processes that span multiple business units.
- Identify services to support these business processes.
- Identify operations from existing systems that could expose as services to support these business processes.
- Identify common supporting infrastructure services.
- Review the process undertaken and incrementally expand by including more business processes (and services).
- Build an application catalogue for future reuse to reduce cost of development.

6. Conclusion

SOA requires enterprises to identify the services infrastructure to deliver the required business solutions. Although SOA promises huge gains as it is based on sound principles of coarse-grained, loosely coupled, standards-based, interoperable, reusable services, there are also numerous challenges such as requirement of a change in mind set, huge initial investment, unreliability of Internet protocols, evolving standards and the newness of the approach. SOA is an effective paradigm for enterprise application integration and therefore enterprises need to be planning for it. They also need to be aware of the vendor hype and be extra vigilant when committing huge sums of money in a technology that is still evolving.

In this paper, we have discussed the relevance of SOA approach to modern businesses. We have outlined the benefits and inherent issues of SOA and reviewed the relevant technologies and tool kits recently released by well-known vendors – noting that a majority of these tools are difficult to use and do not necessarily deliver the benefits they claim to offer. The next generation tools will, hopefully, provide better facilities for testing and improved support for monitoring and management. Guidance on building and implementation of SOA has also been given. The objective is to provide some useful background information for enterprises wishing to embark on the road to SOA.

References:

[1] Cartwright I and Doernenburg E, Time to jump on the SOA bandwagon, *IT Now, British Computer Society*, May 2006

[2] Knorr E and Rist O, 10 Steps to SOA, *Info World - San Mateo*, Vol. 27, Issue 45, pp 23-35, Nov 2005

[3] Barnes D, The service oriented architecture: more than just another TLA?, *BCS, UK*.
Retrieved Feb 2007, from
www.bcs.org/server.php?show= ConWebDoc.3041

[4] Kodali R R, What is service oriented architecture? *JavaWorld.com*, 13 June 2005, Retrieved Feb 2007, from
http://www.javaworld.com/javaworld/jw-06-2005/

[5] Savvas A, SOA forecast to be a key trend in 2007, Jan 2007.
Retrieved Apr 2007, from
www.computerweekly.com/Articles/2007/
01/12/221137/soa-forecast-to-be-a-key-trend-in-2007.htm

[6] Sonic Software Solutions, Service oriented architecture.
Retrieved Feb 2007, from
http://www.sonicsoftware.com/solutions

[7] Johnson, B, The benefits of service oriented architecture, *Objectsharp Consulting.*
Retrieved March 2007, from
http://objectsharp.com/cs/blogs/ bruce/pages/235.asp

[8] Mahmood Z, Service oriented architecture: Potential benefits and challenges, *Proc. WSEAS Int. Conf., Crete, Greece*, July 2007

[9] Overall D, Have we been there before?, *Opinions, Computer Weekly, UK*, 11 April 2006

[10] Fowler M, Patterns of enterprise application architecture, *Addison Wesley*, 2002

[11] Groves D, Successfully planning for SOA, *BEA Systems Worldwide*, 11 Sept 2005

[12] Zimmermann O, Krogdahl P, Gee C, Elements of service-oriented analysis and design, *IBM Corporation*, 2 June 2004

[13] Hohpe G, Stairway to heaven, *Software Development*, May 2002

[14] Hohpe G, Developing software in a service-oriented world, *Whitepaper, ThoughtWorks Inc.*, Jan 2005.

[15] Mahmoud Q H, Service-oriented architecture and web services: the road to enterprise application integration, *Sun Microsystems*, Apr 2005

[16] Ward-Dutton N, The SOA tool pyramid, March 2007.
Retrieved Apr 2007, from
http://www.mwdadvisors.com/blog/2007/03/soa-tool-pyramid.html

[17] Colan M, Service-oriented architecture expands the vision of web services – part1, *IBM Corporation*, 21 April 2004.

[18] BEA SOA360.
Retrieved Apr 07, from
www.bea.com/framework.jsp?CNT=soa360.jsp&FP=/content

[19] The eclispe SOA Tools Platform Project.
Retrieved Apr 2007, from
www.eclipse.org/stp/

[20] Bharti N, IBM releases WSDM tools for SOA management, Dec 2005.
Retrieved Apr 2007, from
www.theserverside.com/news/thread.tss?thread_id=38084

[21] Oasis, An introduction to WSDM – Committee draft, Feb 2006.
Retrieved Apr 2007, from
http://www.oasisopen.org/committees/
tc_home.php?wg_abbrev=wsdm

[22] CapeClear Software.
Retrieved Apr 2007, from
http://www.capeclear.com

[23] The CapeClear ESNB Platform.
Retrieved Apr 2007, from
http://developer.capeclear.com/7.0.0/help/
index.jsp?topic=/com
.capeclear.eclipse.studiohelp/server/svroverview-thecapeclearesbplatform.html

[24] Boulton C, Cape Clear crafts SOA tool, March 2004.
Retrieved Apr 2007, from
http://www.internetnews.com/dev-news/

[25] Darryl K T, Stampede Rolls Out New Tools for SOA, Apr 06.
Retrieved Apr 07, from
www. eweek.com/article2/0,1895,1953591,00.asp