Service Orientated Architecture Primer

What is SOA?

Gartner believes that SOA is changing software [2]. This paper outlines the concepts behind SOA and shows where these ideas and supporting technologies are heading. As discussed in [1], architecture's goal is to hide complexity. Service Orientated Architecture (SOA) is primarily a design philosophy that achieves this by abstracting a system into a series of communicating business processes (services) that are connected via a network.

The focus on service provides some clear blue water between SOA and the last fashionable integration technology, object buses (e.g. CORBA). Object buses allowed architectures to be built from a set of cooperating distributed objects, with focus on entities, and usually small ones. SOA, by contrast, focuses on services (e.g. enter deal) rather than entities (e.g. deal).

Services provide a coarse grained interface model. The pendulum has swung from large monolithic applications, to distributed objects and has now settled somewhere in the middle. There are a few of key reasons why the current position is an improvement:

1. Services map better to business processes than data entities since the focus is on activities. This can improve the dialogue between the business and IT since the business can think of the architecture as a suite of co-operating business processes.
2. Coarse grained interfaces provide a simpler way of exposing large systems such as SAP. Instead of decomposing a set of SAP APIs into entities, they are composed into a set of services representing business processes.
3. SOA has an implicit focus on simplicity, SOA are generally based on mature technologies with transports that are, in general, quasi-human readable. This makes debugging, troubleshooting and construction simpler and more cost effective.
4. SOA's focus is linkage of disparate systems in heterogeneous environments. The protocols and technology used tend to have broad platform support or are entirely standards based.
5. B2B initiatives were based on EDI. EDI formats were usually complex, based on different meta-models and required specialist support. SOAs tend to incorporate XML data content so B2B standard messages can be represented using the standard XML meta-model and manipulated using standard tools.
6. XML usage. The focus on XML provides the architect with a standard set of tools for defining, validating, transforming and referencing message content.

7. Services are interface based, forcing architectures to move up a level of abstraction. Key is not interfacing to ERP but, for example, interfacing to Customer Services.

In SOA there are three core entities:

- *Service Providers*: Provides an interface to a component that performs a set of tasks.
- *Service Requesters*: Discovers and invokes actions on a Service Provider.
- *Service Brokers*: Repositories for interfaces published by Service Providers.

Currently many SOA based architectures do not use a Service Broker, preferring private discovery of services. However, as SOA becomes more established expect interface repositories to become more prevalent.

Two Approaches

SOAs popularity has primarily been driven by the current interest in web services. Web services implement SOA using HTTP and SOAP[9]. Simplistically, SOAP is an XML based protocol that wraps an XML element in an envelope, so it is ready for transmission. SOAP is associated with a large number of standards for functions such as interface descriptions (WSDL)[5] and service discovery (UDDI)[13]. The initial problem with SOAP based SOA was the lack of standards addressing transactions, security, workflow and business process integration etc. Recently standards have appeared for these key areas, details of which can be found in, amongst many places, [11], [12] and [14]. Unfortunately there are two different groups of players in the standards marketplace, Microsoft and IBM on one side and Sun and ORACLE on the other. Until the competition over standards definition settles down, the vendors your organisation is engaged with will likely inform your standards choice. The number of standards can be daunting. One organisation, WS-I (Web Services Interoperability) seeks to assist in this area by providing “guidelines, conventions and best practices”[10] for the production of interoperable web services.

The web service model described above is one approach but SOA is about design more than technology [6]. SOA can be built on more mature technologies; one approach combines XML message content (often in the form of SOAP) with message-orientated middleware (MOM) such as MQ or TIBCO. MOM and supporting technologies provides support for transactions, sequencing, workflow and transformations. Furthermore popular MOM tends to be mature and available on a wide variety of platforms.

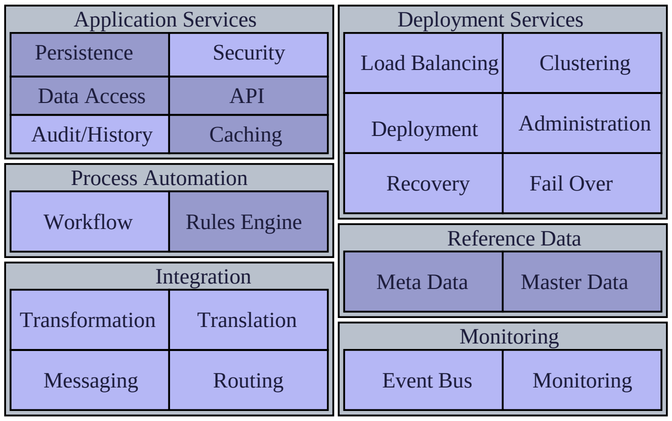
In the short term enterprises have been adopting MOM based SOA as a preferred model, particularly in organisations with existing MOM. However, as the web service technologies mature, some organisations will certainly move to a web service based approach to SOA implementations.

Inherent Complexity

Despite the hype surrounding SOA, raw SOA currently only addresses mechanisms for structuring architecture along behavioural lines. It is important to realise that SOA itself does not specifically address many of the common problems associated with architecting systems:

- Semantics and ontologies.
- Reference Data (Master Data).
- Business rule support.
- Transaction management.
- Orchestration, workflow and sequencing.
- Security.
- Performance.
- Scalability.
- Availability.
- Disaster Recovery.
- Audit and History,
- Distribution policy (centralised or geographically distributed).
- Operability.
- Message content, e.g. standard message formats.
- Governance.

The good news is that many vendors of EAI enabling technology address some of the above issues in their SOA product set as part of providing web service interoperability. Furthermore, XML, messaging, web services and application servers are converging and the result is being called the Enterprise Service Bus (ESB). The ESB aims to provide not only provide a SOA model, but to also provide a standards based platform on which many of the above issues can be addressed [3]. Examining the types of features required in business systems provides us with the information to draw a high level reference architecture (see below).



A large percentage of this reference architecture (shaded light blue) is, to a greater or lesser extent, the target of the current activity in the ESB space. ESB vendors are targeting the current integration vendors, and are aiming specifically for lower integration costs, primarily by providing more sophisticated plumbing and reducing the need for large scale consultancy efforts. EBS is the first step from SOA to “Next Generation Application Integration”.

Success with SOA

To be successful with SOA requires many non-technological activities to come together. Firstly, an organisation needs a service governance model. With many clients accessing a service it is important to understand how to manage the evolution of the service over time. Secondly, the focus needs to remain on the business; deployment of SOA based architecture should revolve around satisfying identified needs within the organisation. Thirdly, service design is paramount. Services need to address business processes. If the level of abstraction is not raised then the organisation becomes filled with numerous small services, resulting in duplication and high configuration and maintenance costs. Last, but most importantly, a successful SOA architecture needs to reside within some form of enterprise architecture framework and roadmap [7] [8].

Conclusion

SOA provides an architectural model that is business process orientated. It supports heterogeneous environments and raises the level of abstraction within EAI architectures. However, it is important to view SOA in perspective. SOA does not solve all the problems in EAI and many hard problems lie in wait for the SOA adopter. Furthermore, the convergence of a number of technologies, not least XML, messaging, web services

and application servers, has resulted in a composite technology, the Enterprise Service Bus. An ESB provides a backbone on which to build SOA and should provide many of the underlying building blocks. ESB is still very much in the early adopters stage and it would be prudent to follow the evolution of the technology over the next 12 months. However, it is important to realise that fundamentally SOA is a design philosophy and it should be primarily evaluated in that context.

Bibliography & References

[1] Schneider, A. *Why Architecture?*. http://www.bjss.co.uk/fow

[2] Schulte, R W. *Predicts 2003: SOA Is Changing Software*. Gartner.

[3] Craggs, Steve. *Best of breed ESBs*.

[4] Linthicum, D. *Next Generation Application Integration*. Addison-Wesley.

[5] WSDL: http://www.w3.org/TR/wsdl

[6] Wilkes, L. *SOA and web services*. http://www.looselycoupled.com/opinion/2003/wilkes-soa-infr0415.html

[7] Armour, F., S. Kaisier, S. Y. Lieu. *A big picture look at Enterprise Architectures*. IEEE IT Professional Jan/Feb 1999.

[8] Smith, D., L. O'Brien, M. Barbacci, F. Coallier. *A Roadmap for Enterprise Integration*. Proceedings of the 10th International Workshop on Software Technology and Engineering Practice.

[9] W3C XML Protocol Working Group. http://www.w3.org/2000/xp/Group/

[10]WS-I. http://www.ws-i.org/Documents.aspx

[11]Transaction Specification Index Page. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/wsatspecindex.asp

[12]Business Process Execution Language. http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/

[13]UDDI. http://www.uddi.org/

[14]OASIS. http://www.oasis-open.org/