Dunelm
Services Limited

Consultants in Applied Research & Development for
Information & Communications Technology

Session 4: Enterprise Integration

Service Oriented Architecture: Preparing Your Business & IT Products

2009 © Dunelm Services Limited

Publication Information

The publication history for this booklet is:

Version	*Date*	*Summary of Amendments*
A	June, 2009	The first release of this booklet.

License

This material is made available under a Creative Commons License. This License, termed 'UK: England and Wales Attribution version 2.0', enables anyone to copy, distribute, display and perform the work, and to make derivative works. The only condition is that attribution credit must be made to the original author; details for how this credit must be given are available on the Dunelm web-site at www.dunelm.com/projects and following the links for the Digital 20/20 project. Further details on the Creative Commons License are at: creativecommons.org/licenses/by/2.0/uk/

Executive Summary

One of the main drivers for Service Oriented Architecture (SOA) is the increasing need to address interoperability in Enterprise systems and that is interoperability across many systems and many sites; this will almost certainly include integration with external organisations e.g. suppliers, funders, etc. Traditionally this is addressed using Enterprise Architecture Frameworks. SOA should be one of the possible implementation architectures for the Enterprise Architecture. Enterprise Architecture is concerned with ensuring that the business processes of an organisation are integrated irrespective of how those processes are implemented and located across the organisation.

In this fourth session of the SOA series the relationship between SOA and Enterprise Integration as a whole will be discussed. While the focus is on Enterprise Integration the discussion addresses developing, using and supplying products that can be used in such environments.

Table of Contents

Publication Information 2
Executive Summary 3
1. Introduction 5
2. Enterprise Integration 7
3. EA Frameworks 10
4. Addressing Scale & Complexity 13
5. IT/SOA Governance 16
6. Evolution from a Legacy System 19
7. Increasing SOA Complexity 22
8. Repositories & Registries 26
9. Enterprise Service Bus 27
10. Enterprise-Capable Products 30
11. Enterprise-Capable Systems 33
12. Typical EAI Scenarios 35
13. Avoiding Deployment Problems 39
14. In Conclusion 42
Appendix A – Bibliography 45
Appendix B – Acronyms 46
Index 48

1. Introduction

In this fourth session, the relationship between SOA and Enterprise Integration as a whole is discussed. While the focus is on Enterprise Integration the discussion addresses developing, using and supplying products that can be used in such environments. Enterprise Integration is a technical field of Enterprise Architecture. The definition for Enterprise Architecture that we will use is taken from MIT and is “the organising logic for business processes and IT infrastructure reflecting the integration and standardization requirements of an organisation’s operating mode”. Enterprise Architecture describes enterprise applications and systems with their relationships to enterprise business goals. SOA is one way for how Enterprise Integration can be achieved.

The three learning objectives for this Session are to:

- Explain how SOA fits into broader Enterprise Architecture frameworks and how legacy systems can be made SOA compatible. Some market sectors, such as defence, have created their own Enterprise Architecture frameworks. These frameworks are used to provide the language for procurement, implementation and deployment of systems in that sector;
- Address the business facing issues including governance, business and technology migration, for SOA-capable systems and products. SOA-based systems must themselves be

managed, using agreed SOA governance processes and best practice for deploying such systems should involve a clear migration path from the current to the new architecture;

- Discuss some typical scenarios encountered with Enterprise Integration. We look at the characteristics that constitute enterprise-capable products and systems, typical scenarios well suited to enterprise architecture integration, and identify the typical deployment problems and how they should be avoided.

The material covered in this Session starts with a review of Enterprise Integration and its relationship to Enterprise Architecture. Enterprise systems tend to be large and complex and so scalability for any component is always a concern. This leads onto the management of SOA solutions in the enterprise and of IT and SOA Governance. Most enterprise solutions grow from humbler origins and so architects must address system evolution and increasing complexity. Next we look at the core technology that is required to support complex large-scale systems and includes the need for federated repositories and registries, and the Enterprise Service Bus (ESB). We then discuss the issues to be addressed when designing enterprise-capable products or supporting enterprise-capable systems and describe some typical scenarios. Finally some of the problems encountered in deploying enterprise solutions are discussed.

2. Enterprise Integration

When undertaking enterprise integration the issues to be addressed are:

- Rationalisation of multiple servers and systems – the number of physical resources can be reduced by adopting virtualisation in its many forms e.g. server, network, database, etc. Further reduction can be achieved by removing unnecessary duplication of physical and logical resources. For example, this would include the use of single sign-on for access to all resources. Older resources can now be used to provide redundancy, fault tolerance and off-site failure recovery capability. When designing such a system there are trade-offs between centralisation versus distribution of information and the requirement to protect data integrity, availability and accuracy;

- Interconnection of geographically distributed sites using a variety of networking technologies. The use of the Internet and the Web has removed exposure to the old ad-hoc telecommunication interoperability agreements that were required for systems such as X.25. However, the Internet is not the same everywhere. For example, some parts are already IPv6 savvy whereas others still support IPv4 only. Internetworking requires support for wireless and mobile devices, as well as roaming access for virtual private network connections. For larger enterprises, use of enterprise-wide virtual private networks may be cost effective, especially if there are strict performance requirements that the use of the general Internet cannot provide;

- Maintain heterogeneity and avoid imposing homogeneity. Different parts of an organisation will have their own, legitimate, work practices and system requirements. A focus on interoperability as opposed to uniformity should permit these differences. Availability of information and services does not imply uniformity, or homogeneity. There is a tension between permitting heterogeneous systems and controlling maintenance costs but in large organisations each site may have their own small support team for specialist systems and applications;
- Ensure architectural, service and process flexibility so that changes can be easily implemented and deployed. Speed of ability to make operational changes due to changing business demands is of increasing importance. Deploying changes should be possible for a single point of management control, which may itself vary depending on the nature of the changes. A services approach coupled with workflow will allow new business processes to be deployed quickly and easily;
- Support for all of the regulatory and audit requirements imposed by the various legislative frameworks in which the organisation operates. The implication is that the relevant information from the various sources has to be collected, collated and analysed at different points across the enterprise. Different styles of reports will be required and these too must be available to suitably authorised users.

- Optimisation of the business processes. The IT systems should simplify, as much as possible, the processes undertaken by the business i.e. make them as simple as possible. Over time, every business process should be improved, irrespective of whether it is manual or computerised. For processes that have varying resource demands, it may not be commercially sensible to support more than 70-80% of those resources. The missing 20% could be supplied using externally supplied on-demand services. The external services should be integrated with the internal provision so that users get a seamless service;

- Accessibility and personalisation. Legal requirements are such that systems, applications and content must be readily accessible i.e. usable for the blind, deaf etc. (note that we are all subject to accessibility variations due to screen size, lighting conditions and background noise). Personalisation is the next stage in which the actual information presented as a well as how it is presented can be tailored to meet the specific needs of each user. Include in this is internationalisation whereby alternative screen layouts will be required to support different languages;

- Ensure the supply chain relationships. Every organisation has suppliers, partners and customers. The same organisation will be a different part in many supply chains. Whenever possible, these supply chains should be electronically integrated with the organisation. Quality of service levels, to and from the organisation, can then be monitored to optimise the performance of the organisation as a whole.

3. EA Frameworks

Enterprise architects use various business methods and tools to understand and document the structure of an enterprise. In doing so, they produce documents and models, together called artefacts. These artefacts describe the logical organisation of business strategies, metrics, business capabilities, business processes, information resources, business systems, and networking infrastructure within the enterprise. An enterprise architecture framework is a collection of tools, process models and guidance used by architects to assist in the production of organisation-specific architectural descriptions.

Enterprise architecture has become a key component of the information technology governance process in many organisations. These companies have implemented a formal enterprise architecture process as part of their IT management strategy. While this may imply that enterprise architecture is closely tied to IT, it should be viewed in the broader context of business optimisation in that it addresses business architecture, performance management and process architecture as well as more technical subjects. Depending on the organisation, enterprise architecture teams may also be responsible for some aspects of performance engineering, IT portfolio management and metadata management.

One example of enterprise architecture is the Federal Enterprise Architecture (FEA). This is a three-layered approach consisting of the Enterprise Architecture, Segment Architecture and

Solution Architecture as shown in Figure 1. The Solution Architecture is where the detailed technology implementation is addressed whereas the Enterprise Architecture is focused on the business process strategy. SOA is primarily addressed as part of the Solutions Architecture and the needs of users and developers while supporting the top-level strategic business processes.

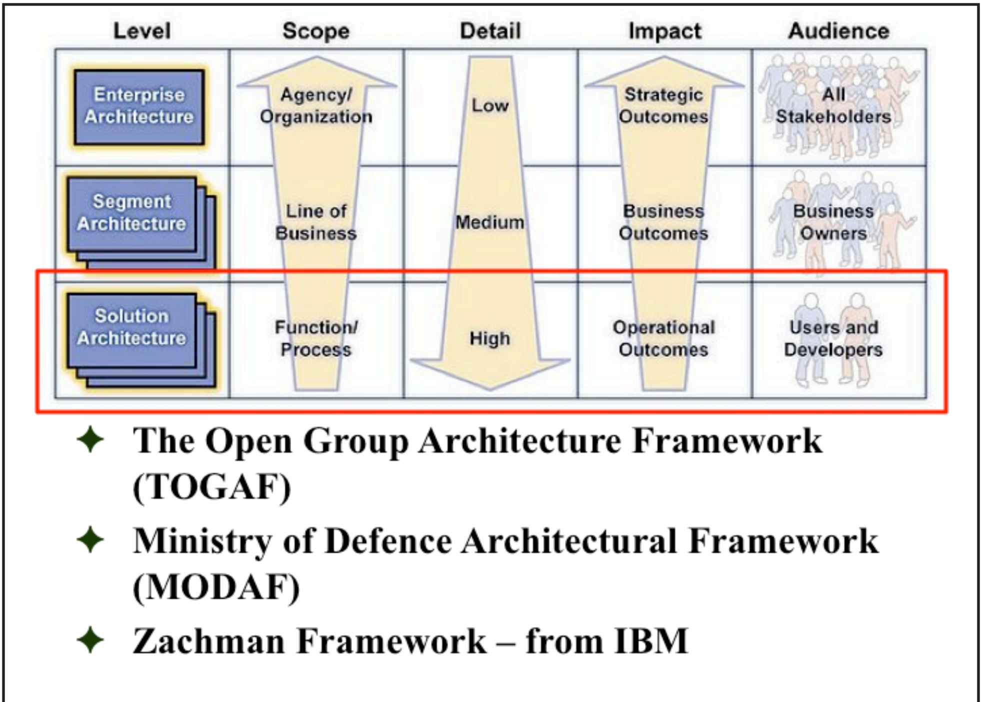

Figure 1 Enterprise architecture frameworks.

Other enterprise architecture frameworks are the Ministry of Defence Architectural Framework (MODAF) [MODAF, 08], The Open Group Architecture Framework (TOGAF) [TOGAF, 09] and the Zachman Framework.

TOGAF is a framework for enterprise architecture with a comprehensive approach to the design, planning, implementation and governance of an enterprise information architecture. The architecture is typically modelled at four levels or domains; Business, Application, Data, Technology. A set of foundation architectures is provided to enable the architecture team to envision the current and future state of the architecture.

4. Addressing Scale & Complexity

By implication, enterprise integration must address the issues of scale and complexity. Enterprise systems tend to be large and complex, both of which create considerable problems. Consider an organization such as IBM. In 2004 IBM had approximately 329,000 employees, worldwide and conducted business in 170 countries. Headquartered in Armonk, New York, USA, turnover was $96.2 billion with $8.4 billion net income. In 2004, measured by revenue, IBM was the biggest provider of IT services ($43B), hardware ($28B) and rental and financing ($3B) and second in software ($15B). Apart from services and manufacturing, it also had several of the World's leading research centres. Now consider supplying and supporting the email service within such an organisation, ensuring that payroll occurs in a timely fashion and providing a set of consolidated financial and management reports. Clearly, this is more challenging than for your average company (most of which are in the small category). IBM requires enterprise integration in every sense of the word.

When addressing scale and complexity in enterprise integration some starting points for technology requirements are:

- Anywhere and anytime access – this begins with providing on-site and off-site access for desktop and mobile devices including laptops and PDAs. Remote access should require use of a corporate virtual private network so that all communication is secure. In more extreme circumstances, the user's environment will be delivered to them irrespective of how they access the system taking into account the access device and the user preferences. This allows a user to easily access the organisation's infrastructure without requiring everyone to have fixed account configurations for each system in the organisation;

- Finding resources – the bigger the organisation, the greater the difficulty in finding the required resources. Repositories should be federated so that any information request automatically searches the full set of physical resources. Directories for information and services need to be made available and accurately maintained. Local caching of information is recommended so that the load on the network infrastructure is minimised;
- System administration and maintenance – the stability of the IT infrastructure is heavily dependent on the quality of the system administration and maintenance teams and procedures. Well designed processes and talented teams prevent problems occurring in the first place. Global support requires a global approach with several teams complementing each other in various time zones. Out-sourcing is also important but this has to be well managed and requires the out-sourcing organisation to have good processes and practices already established;
- Management reporting – the IT infrastructure must provide

the senior management team with accurate and timely information about all aspects of the organisation, including the IT provision itself. This requires layering of information. Summary and detailed information should be separated but available for review as and when required. Multimedia reporting is required as well as standard support for online and hybrid meetings;

- Service availability and reliability – failure of the IT system can be catastrophic for an organisation (the statistics show that over 90% of organisations that suffer catastrophic failure cease to trade within the following two years). On a day-to-day basis, systems need have between 0.99999 and 0.999999 availability i.e. between 4 and 5 nines. Availability of five nines means that the system is unavailable for approximately 1 minute in 70 days (note that an availability of 0.99 implies loss of service for 3.6 days each year). The greater the availability requirement the more expensive the infrastructure because more fault tolerance has to be provided. The service level agreements need to be clear on the level of availability and the system infrastructure designed appropriately;
- The law of 'unintended consequences' – no matter how well the system has been designed there will be unexpected behaviours, some of which will cause problems. Preventative maintenance and effective help desk support will minimise the consequences but, for an international organisation, these must to operate on a 24 hour, seven days a week basis.

5. IT/SOA Governance

“Governance” means the set of practices, policies and processes that are put in place to increase the likelihood that an entity functions as intended. Corporate governance is the set of practices and policies that dictate the way a corporation is controlled and directed, while IT Governance is the set of practices and policies that ensure that IT investments are generating the intended value for the organization. SOA Governance is the set of practices and policies that ensure service oriented architecture and its component services represent and conform to the objectives and requirements of the institution. It involves not only IT support processes to ensure regulatory compliance and quality of service, but also organisational processes to ensure that stakeholders are receiving value by adopting SOA. Figure 2 shows the relationship between these three types of governance.

New service-oriented projects are often undertaken without explicit SOA governance in place. This works if the scope of the project is small, but, as the architecture grows to include new service providers and consumers, the overall reliability and predictability of the architecture begins to decline; this is typically the result of a lack of governance. For example, a lack of attention to IT service boundaries may have resulted in duplicating some old IT service functionality in a new IT service. Or perhaps the demands placed on a certain IT service may have increased dramatically because of a new service consumer, and this may have been unaccounted for – resulting in

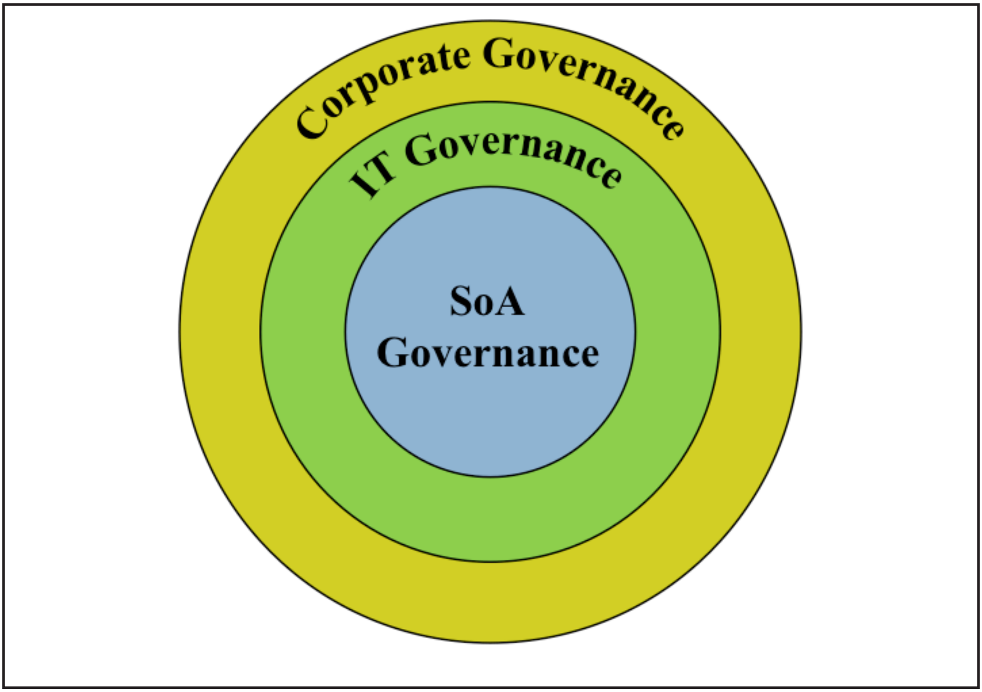

Figure 2 SOA governance.

the inability of the IT service to meet its service level agreement. Furthermore, as enterprise services evolve and new IT service versions are deployed, careful consideration must be taken each time to assess impacts of the changes to other IT services and provide backward-compatibility

In order to prevent many of these potential problems from occurring, some amount of formal governance is critical even in a new service-oriented project to ensure delivery of consistent and reliable IT services. Fortunately, one does not have to adopt all elements of a full-fledged SOA governance solution to gain many of its benefits, and instead can concentrate on using appropriate parts of governance to gain maximum benefit depending on scope and nature of the individual project at hand.

In many cases, there exists some form of governance over the architecture, although it is often informal. More often than not, informal governance is inconsistently applied, usually proving to be insufficient as the scope of the architecture expands. In order to ensure that the goals of the SOA are understood and achieved, some amount of formal governance should be practiced.

However, what amount of governance is the right amount? Some approaches dictate heavy governance from the outset, before there is much chance of most of it doing any good. Many simple situations may require only an assigned person to check periodically for common issues but as the architecture matures, more heavy-weight governance is usually needed. Ideally, an organisation only practices as much SOA governance as is necessary to handle the architecture in its current state and allow for flexible growth and smooth operation into the future. Therefore the goal is to roll out SOA governance practices to keep up with the state of the architecture, which is a moving target.

6. Evolution from a Legacy System

In Session 3, we explored the use of SOA maturity models. The Open Group's Service Integration Maturity Model (OSIMM) identified seven levels of maturity [OSIMM, 06]. Another way of viewing this model is to consider how maturity is increased, especially when evolution from a legacy is required. This evolution has three steps:

- Publish – to make the corporate information available in a standardised format e.g. XML. This includes enterprise-wide agreement on the semantic definitions of the information fields. At this point the data is only available to the hosting system;
- Expose – to make the corporate information available to the enterprise as a whole. This includes addressing information interoperability so that systems can publish and consume the information. This culminates in being able to publish the data through a set of standard service interfaces;
- Compose – to turn the corporate business processes into a set of services that can be combined in different ways to create new processes and capabilities.

A more detailed breakdown, mirroring the seven layers of the OSIMM and reflecting the changes required to move between each level of maturity, is:

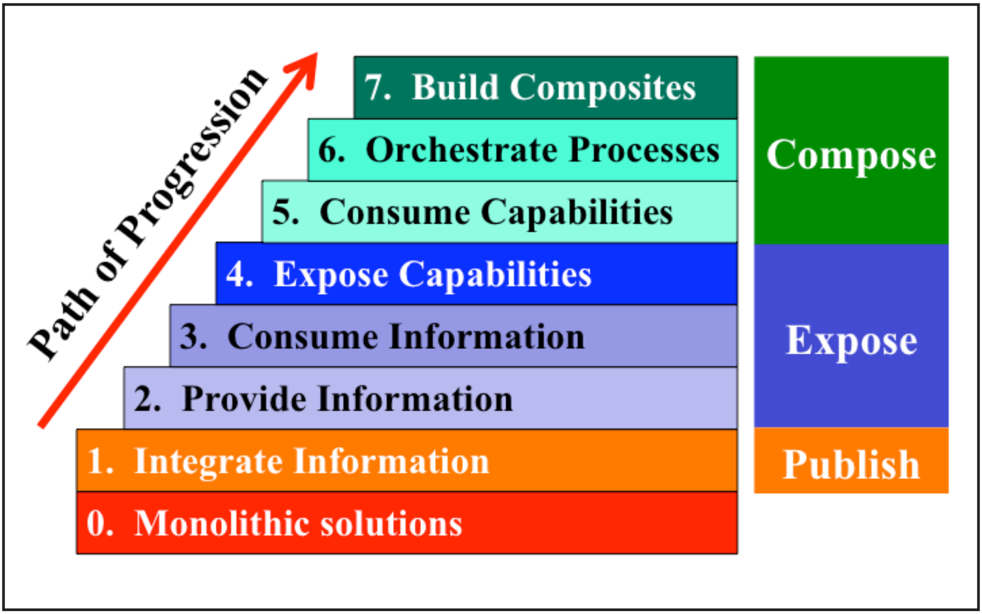

Figure 3 Evolution from a legacy system.

- Integrate information (level 1) – this first step is to look at the monolithic system architecture and to identify the set of data structures. These data structures need to be revealed in standardised formats so that physical exchange becomes possible. This makes it easier to publish data and minimises the number of tools that are required to read that data;
- Provide information – the next step is to make it easier to obtain the data. This involves making it possible for external systems to obtain the information either through a Web server or some other download capability;
- Consume information – this is the process by which local and externally available information can be combined and

used in the local system. Information can now be used to drive events in systems other than that which created the information in the first place;

- Expose capabilities – this is the first stage in turning the capability of an application into an externally available service. Once these services are externally available a single instance of the service can support any number of remote users;

- Consume capabilities – a service can now use other services to provide both the required information and remote functionality. For example, one service could provide a data set to another service so that it can undertake specific processing of that data and will return the new information;

- Orchestrate processes – a business can now start to control how services are co-ordinated to support particular business processes. Each business process is expected to be self contained but will undertake operational responsibilities that are enterprise-wide;

- Build composite solutions (level 7) – at this final level of maturity, individual business processes can be combined to provide increasing complex business solutions. At this level of capability the business is continually evaluating and optimising its processes.

It is important to remember that this evolution may apply to only a subset of an organisation's business processes. The same sequence of evolution is undertaken.

7. Increasing SOA Complexity

In most environments, deployment of services is a step-by-step process. At the most basic breakdown there are three clear stages of evolution for deploying services for an organisation: the single service within a department; multiple services across the organisation; and an inventory of services for intra and inter organisation usage. Figure 4 shows the relationship between these three steps. To explain these stages we'll use a simple example to show the evolution of the deployment.

Consider the IT maintenance team who keep a number of logs about set of resources deployed in the organization. These logs are made available as a maintenance report service that anyone in the IT administration department can access. The service has been designed such that all of the maintenance tools automatically log a change in state of a resource in the log. The service provides management reports that contain summary information and on-demand detailed reports. From an SOA governance perspective the maintenance team will have defined and deployed the service. By definition the IT administration department have ownership and as such are responsible for funding its provision. This is our single service for a single department within an organisation.

In the second step, the IT department decides to allow other departments to use the service; in order to ensure that the service plays well with the services of other departments, the IT department. In order to maximize interoperability and ease

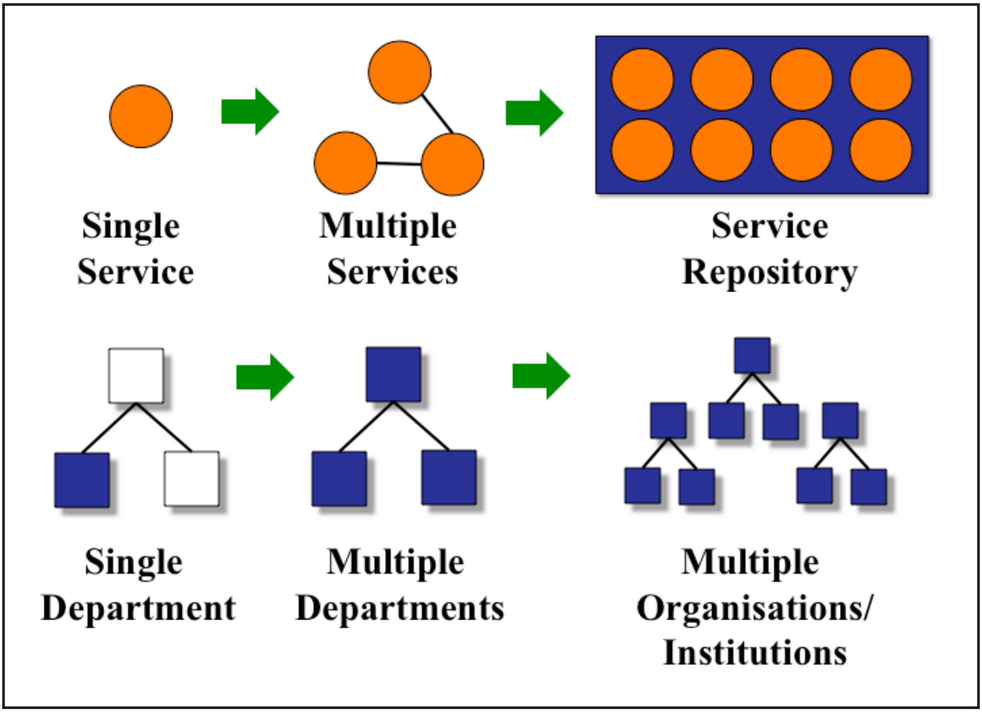

Figure 4 Increasing SOA complexity.

of adoption, the organisation employs standard data formats for its services. The IT department decides to expose more sophisticated capabilities; the users will be allowed to undertake analysis of the logs as opposed to just reading them. In order to support this new level of service requirement, the department starts to implement formal policy conformance auditing for the service. The department intends to automate this as the number of services grow, but decide that it is sufficient to perform the checks manually at this time. The success of this service leads the IT department to create several other services that allow the work of the department to be followed. The senior IT managers now decide to establish a services governance board with representatives from relevant departments around the organisation. The governance board decides that the security policy should enforce access controls to these new services. Other departments now request that some of their business processes are realised as services. We now have the basis for the delivery of multiple services to multiple departments within the organization.

The services are becoming robust and the organisation as a whole is getting value from the approach. The organisation decides that some key external partners will also need to access some of the services. With this new, broader user base, the organisation decides to place overall quality of service requirements on services, including establishing downtime limits during new version deployments. They also recognize that these service endpoints may change over time, so a service registry is created and discoverability requirements are imposed on all services to ensure that users can find the services.

With quality of service rules in place, the architecture has reached a point now where the services may be choreographed to execute business processes. This step creates a need for additional control over the lifecycle and state of services, so the institution adds management capabilities to the services and places management requirements on future services. After seeing success with choreography within the institution, the institution decides to combine its services with other institutions. In order to help assess business value internally and to the other organisations, Key Performance Indicator (KPI) capabilities are identified and monitored. This information is used to optimise the service delivery.

The initial deployment of a single service for use within a single department has now grown to a full SOA solution. The evolution was driven by business needs and built upon incremental success with clear benefits to the organisation provided at each stage of development.

8. Repositories & Registries

In an enterprise system, finding the right information, service, etc. is one of the more important but difficult activities. The answer is to use registries and repositories. A registry is used to store information about resources, normally in the form of metadata, whereas a repository will contain the resources themselves. The registry will always provide the location of the actual resource. Access to registries and repositories requires an appropriate protocol cf. accessing an electronic library using the Search/Retrieve Web Service. In many cases several registries and repositories will be maintained to form a federated architecture. Therefore the access protocols should use federated search, etc. so that the user does not have to search each repository/registry individually.

For services, a repository should store the web services description language, XML schema definition and the business process execution language files that describe the services (more about these files in Session 6). Other important information is the service policies (including service level agreements) and various metadata files. Many of these files are not for human consumption but they enable computerised systems to make the services available to users.

9. Enterprise Service Bus

Enterprise Service Bus (ESB) refers to a software architecture construct typically implemented by technologies found as middleware infrastructure products, usually based on recognized standards, which provide fundamental services for complex architectures via an event-driven and standards-based messaging engine (the bus). An ESB generally provides an abstraction layer on top of an implementation of an enterprise messaging system, which allows integration architects to exploit the value of messaging without writing code. Contrary to the more classical enterprise application integration (EAI) approach of a monolithic stack in a hub and spoke architecture, the foundation of an enterprise service bus is built of base functions broken up into their constituent parts, with distributed deployment where needed, working in harmony as necessary.

An ESB does not implement a SOA but provides the features with which one may be implemented. Although it is a common belief, an ESB is not necessarily Web Services based. An ESB should be standards-based and flexible, supporting many transport mediums, capable of implementing both traditional SOA patterns as well as SOA 2.0-enriched business architectures. ESBs attempt to isolate the coupling between the service called and the transport medium. Most ESB providers incorporate SOA principles and allow for independent message format.

There is some disagreement on whether an enterprise service bus is an architectural style, a software product, or a group of software products. While use of an ESB certainly implies adherence to a particular architecture, the term “enterprise service bus” is almost always used to denote the software infrastructure that enables such architectures. An ESB brings flow-related concepts such as transformation and routing to SOA. An ESB can also provide an abstraction for endpoints. This promotes flexibility in the transport layer and enables loose coupling and easy connection between services.

In such a complex architecture, the ESB is the piece of software that lies between the business applications and enables communication among them. Ideally, the ESB should be able to replace all direct contact with the applications on the bus, so that all communication takes place via the bus. In order to achieve this objective, the bus must encapsulate the functionality offered by its component applications in a meaningful way. This is typically accomplished through the use of an enterprise message model. The message model defines a standard set of messages that the ESB will both transmit and receive. When the ESB receives a message, it routes the message to the appropriate application. Often, because the application was not built with the message model in mind, the ESB will have to transform the message into a legacy format that is understandable by the application. The software responsible for effecting these transformations is referred to as an adapter (analogous to a physical adapter). It is not widely agreed whether these adapters should be considered part of the ESB or not.

The connection between the enterprise message model and the functionality offered by the applications is crucial. If the message model does not completely encapsulate the applications' functionality, then other applications that desire that functionality will be forced to bypass the bus and invoke the applications directly. Doing so violates all of the principles outlined above, and negates many of the advantages of using an ESB.

10. Enterprise-Capable Products

If a decision has been made that a product should be enterprise capable, there are a number of implications for the requirement specification of such a product. Key amongst these is:

- Service definition – the functionality of the product must be made available through a service definition. This includes remote maintenance and support. Desktop and hosted versions should be available with the same user interface. The average user should be unaware of whether they are using a desktop installed version or its hosted equivalent. The service interfaces should have human and machine readable formats and whenever possible software developer kits should be created to enable the development of third party plug-ins;
- Workflow – a workflow engine can be included in the product. This would allow users to construct their own workflows from the perspective of the services supplied by the product. An alternative approach would be to require the use of an enterprise's own workflow engine. The Business Process Execution Language (BPEL) and the many open source BPEL-engines are one common way of supplying workflow support [BPEL, 07]. However, a workflow wizard should be supplied so that users do not have to understand BPEL before being able to create a workflow;
- Authentication, Authorisation and Access control (AAA) –

AAA-based protection is essential. However, it is important that single-sign-on integration is possible. For example, support for the Lightweight Directory Access Protocol should be considered. Proprietary security approaches should be avoided. A log system should be developed so that use, and attempted use, of the product are recorded. This log should be made available to any remotely authorised management system;

- Separation of information from presentation – information has to be available to many different systems and devices. Separation of the presentation format from the data itself is essential. Presentation through standard browsers or using Rich Internet Applications (RIAs) is now a common requirement. The use of widgets, plug-ins, web apps and other similar approaches should be used to make key information readily available;
- Consistent semantic mapping between business rules and information – correct, common interpretation of data is always problematic. For example, even simple dates can be misinterpreted if the differences between European and North American styles are not handled. Each organisation will have a data dictionary. The data storage and the associated business rules should be separated and wizards used for mapping between them.

Interoperability is an essential property. The technology used for the implementation should not limit where, when and how the product is used. A consistent approach to interoperability will remove most of these issues. Usage of something like an ESB will provide a common interoperability implementation technology. An ESB-Adapter can be used to isolate the service from the actual flavour of ESB with new adapters used to link to other flavours of ESB.

11. Enterprise-Capable Systems

Systems that are enterprise-capable have a number of common characteristics that make it easier for enterprise-capable products to be deployed. These characteristics include:

- A clear set of information semantics – this means that the information throughout the organisation has consistent meaning and can be combined by a user confident in this consistency. Consistent meaning is very difficult once an organisation covers more than one country and if more than one market sector is supported. Mapping of equivalent information can be undertaken but this has to be agreed at the start of any integration process. Under no circumstances should end users be required to undertake this mapping because at best it will be done badly and at worst it will not be done;
- Requires and can provide information when and where it is required – timely access to information is essential. Enterprise systems can provide the required information when and where it is required. This information is accurate and consistent with all of the related information distributed across the enterprise. The information is reliably stored with duplication permitted only under controlled conditions. The information can be presented in whatever form is required i.e. it can be delivered to the desk-top, in print form, on PDAs etc.

- Containing layered services that range from the most common to business process specific functionality – this means that the service provision has been rationalised and core common services such as authorisation, authentication, workflow, calendaring, etc. have been separated and structured such that they can be used to support any other service. The services that are more aligned to specific business processes have been designed to reuse the common services. The same approach must be adopted for any new product;
- Flexibility to support almost unconstrained change – the organisation has recognised that it must be prepared to make continual small changes and improvements to remain competitive. Consequently, the underlying IT infrastructure has been designed to make it easy to change the operational business processes and to deliver the required services when and where required. Once again, any new product must be capable of being easily integrated in this service oriented computing approach otherwise it has only limited value to the organisation as a whole.

12. Typical EAI Scenarios

Let's now look at four scenarios in which enterprise architecture integration could or should play an important role.

The first scenario is the support of education within a University or Further Education (FE) college. A typical University in the UK employees thousand of people, supports tens of thousands of students studying hundreds of different courses, participates in extensive research with collaborators located in many countries and has premises that can spread across several countries. Apart from the research aspect, the FE colleges have a broader range of teaching activities but on a smaller scale. Both types of institution have classic enterprise integration requirements that can be especially challenging due to the idiosyncratic nature of the needs of different teachers and learners.

It is essential that the systems within a College and University are integrated otherwise manual intervention will introduce considerable delays and inaccuracies in key information e.g. course lists, institution registration numbers, funding source tracking, human resource management, etc. In some of the larger Universities, there is already a move towards the use of enterprise service bus infrastructures. The cyclic nature of the teaching activities also means that many support administration services have to be repeated at regular intervals and the set of permitted users of the various systems changes substantially.
It is essential that at least the student information aspects are service based and that this information is made available to

a wide range of appropriately authorised systems. Increased government information requirements also demand that the senior management team be provided with more accurate and timely information about all activities within their institution. Again a service-oriented computing infrastructure can make this a reality.

The second scenario looks at just-in-time supply chains using Warehouse Management. Software systems in this market are typically focused on third party logistics (3PL). These systems are used to provide a managed warehouse service. The limitation of many 3PL systems is their lack of enterprise integration with their supply chain. Apart from the high-end systems that support supermarkets (using either EDI or the more recent ebXML), etc. supply chain integration is based upon email exchanges and attached documents. This needs to be replaced with service integration so that the supply chain uses an agreed interoperability architecture based upon the principles of SOA. Any sector agreed set of interoperability standards should be used or developed on behalf of the sector. There is also a need for the suppliers of 3PL system to create a generic platform i.e. a 4PL that could then be supplied as a hosted managed service or supplied as software as a service. This requires a rethink of the underlying software architecture so that service-oriented computing can be used.

The third scenario looks at information integration for Facilities Management. The high-end systems are capable of managing more than one building. As such they collect a huge amount of information about how a facility is used i.e. energy usage, staff access, etc. In general, reporting is supplied using proprietary features in the facilities management system. The first new requirements are to provide workflow integration and enterprise reporting so that the facility users/owners can obtain on-demand information about their facilities and with report formats that reflect their requirements.

The last scenario is the NHS Connecting for Health and its NHS National Programme for IT (NPfIT). NPfIT, is an initiative by the Department of Health in England to move the National Health Service (NHS) towards a single, centrally-mandated electronic care record for patients and to connect tens of thousands of General Practitioners (GPs) to hundreds of hospitals, providing secure and audited access to these records by authorised health professionals. The Department of Health agency NHS Connecting for Health (NHS CFH) is responsible for delivering this programme. NPfIT is said by the NHS CFH agency to be “the world’s biggest civil information technology programme”. This is enterprise integration on the grandest of scales. It has also been a catalogue of disaster and could become a case study of how not to undertake enterprise integration.

Recently released Gateway audit reports show that whilst there was a grand vision this was not translated into practical reality. Many of the key stakeholders and end-users were not involved in the specification of the system and indeed were unconvinced of the need for such a system. Interoperability was not an underlying feature of the architecture with a largely homogeneous system being imposed on the GPs and hospitals. Timescales were unreasonable and the approach spread over too many objectives.

A better approach would have been to focus on small gradual service roll-outs. The services should have been selected in co-operation with the key stakeholders with differences permitted to reflect the needs of the various communities. Service interoperability should have been the starting point so that solutions from many specialist suppliers could have been integrated as opposed to imposing solutions from a few large consortia.

13. Avoiding Deployment Problems

Experience has shown that there are several common problems encountered when deploying SOA-based solutions. These include:

- Management of the system is essential. All of the services deployed in a system must be manageable. If a service cannot be remotely managed then it should not be deployed. Management information should be continuous and proactive i.e. it should be used to anticipate problems instead, of being predominantly reactive. For enterprise systems this is only possible if many of the management functions are automated with suitable alert thresholds used to identify areas of concern. Also note that the management system itself needs to have a back-up and recovery capability;

- System-wide security is notoriously difficult to achieve. Electronic systems are particularly vulnerable to a number of attacks including denial of service. Specialist advice on SOA-based security should be used. Security should be considered during the requirement definition. Similarly, access compliance and intrusion detection should be deployed such that if a system is compromised there will be audit trails to provide guidance on how to improve the security of the system;

- Users perceive the system through an end-to-end perspective i.e. they have no, nor need, any understanding of how a service is delivered. Management systems must also address end-to-end capabilities as well as providing a mechanism by which end-to-end operations are revealed in terms of their composition. A user will perceive failure of an intermediate system as failure of the service itself. Therefore management is a layered activity and in well designed systems end-to-end capabilities should be immune to a certain level of intermediate system failures;

- Flexibility to change as demand changes. The operational requirements in an enterprise system will change continuously. The operational capabilities must be changed appropriately, and in a timely and controlled fashion. Many changes in demand can be anticipated by observing the operational characteristics of the system and understanding the strategic objectives of the organisation as a whole. The former requires the appropriate system monitoring, analysis and response whereas the latter requires the senior IT managers to be exposed to, and briefed on, the decisions made at the corporate Board level;

- Access to the appropriate reports is the only way an organisation can monitor performance. The appropriate information must be supplied in a timely fashion. It must also be possible to combine reports from multiple sources to create new reports. All reports must be archived and readily available. Reports must be accurate and this means they should use information that is less than an hour old. Whenever possible, data processing should be computerised and automated. Finally, access to information needs to be available not only to the managers but also their direct support staff who will help them create the required reports;

- Virtualization of the IT infrastructure is of increasing importance. Adoption of virtualisation at all levels of deployment is recommended i.e. server to desktop. This considerably simplifies service recovery, service maintenance and service improvement.

No matter how well prepared or planned, deployment of new services will always encounter the unexpected. A well governed system will make it easier to overcome most of this problems.

14. In Conclusion

One of the main drivers for SOA is the increasing need to address interoperability in Enterprise systems and that is interoperability across many systems and many sites; this will almost certainly include integration with external organisations e.g. suppliers, funders, etc. Traditionally this is addressed using an appropriate Enterprise Architecture Framework. SOA should be one of the possible implementation architectures for the enterprise architecture. Enterprise architecture is concerned with ensuring that the business processes of an organisation are integrated irrespective of how those processes are implemented and located across the organisation. Enterprise architecture describes enterprise applications and systems with their relationships to enterprise business goals. Enterprise Integration is a technical field of enterprise architecture and SOA is one way to achieve enterprise integration.

By implication, enterprise integration must address the issues of scale and complexity. Enterprise systems tend to be large and complex, both of which create considerable problems. When addressing scale and complexity in enterprise integration some starting points for technology requirements to be considered are: anywhere and anytime access; finding resources; system administration and maintenance; management reporting; service availability and reliability; and the law of ‘unintended consequences’. Management, or more broadly governance, is essential in enterprise systems. “Governance” means the set of practices, policies and processes that are put in place to increase the likelihood that an entity functions as intended. Corporate governance is the set of practices and policies that dictate the way a corporation is controlled and directed, while IT Governance is the set of practices and policies that ensure that IT investments are generating the intended value for the organization. SOA Governance is the set of practices and policies that ensure service oriented architecture and its component services represent and conform to the objectives and requirements of the institution. It involves not only IT support processes to ensure regulatory compliance and quality of service, but also organisational processes to ensure that stakeholders are receiving value by adopting SOA.

In most environments, deployment of services is a step-by-step process. At the most basic breakdown there are three clear stages of evolution for deploying services for an organisation: the single service within a department; multiple services across the organisation; and an inventory of services for intra and inter organisation usage. Many deployment problems can be avoided by careful planning such that: management of the system is implicit is every aspect of the system; system-wide security, notoriously difficult to achieve, must be addressed as part of the requirement specification of the system; users perceive the system through an end-to-end perspective and service failure may in fact be caused by failures in any of the 'hidden' intermediate systems and so a layered management approach is required; there is flexibility to change as demand changes; access to appropriate information is provided as and when required;

there is full use of virtualization.

Enterprise capable products and systems need careful specification and design. Moving towards an SOA-based solution must be planned especially when it involves evolution from a legacy system.

APPENDIX A – BIBLIOGRAPHY

[BPEL, 07] *Web Services Business Process Execution Language Version 2.0*, OASIS Standard, 11th April 2007. [http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html].

[MODAF, 08] *Ministry of Defence Architecture Framework*, v1.2, Ministry of Defence, UK, September 2008. http://www.modaf.org.uk/

[OSIMM, 06] *The Open Group Service Integration Maturity Model (OSIMM)*, Draft, The Open Group, 2006.

[TOGAF, 09] *The Open Group Architecture Framework*, v9, The Open Group, February 2009. http://www.opengroup.org/togaf/

[Zachman] *Zachman Framework*, http://www.zachmaninternational.com/index.php/the-zachman-framework.

Appendix B – Acronyms

3PL	Third Part Logistics
AAA	Authentication, Authorisation and Access
BPEL	Business Process Execution Language
EAI	Enterprise Application Integration
ebXML	Electronic Business XML
EDI	Electronic Data Interchange
ESB	Enterprise Service Bus
FE	Further Education
FEA	Federal Enterprise Architecture
GP	General Practicioner
IP	Internet Protocol
IT	Information Technology
KPI	Key Performance Indicator
MIT	Massachusetts Institute of Technology
MODAF	Ministry of Defence Architecture Framework
NHS	National Health Service
NHS CFH	NHS Connecting for Health

NPfIT	National Programme for IT
OSIMM	Open Group Service Integration Maturity Model
PDA	Personal Digital Assistant
RIA	Rich Internet Application
SOA	Service Oriented Architecture
TOGAF	The Open Group Architecture Framework
UK	United Kingdom
USA	United States of America
XML	Extensible Markup Language

Index

Symbols

3PL 36

A

AAA 30, 31

Authentication, Authorisation and Access. *See* AAA

B

BPEL 30, 45, 46

Business Process Execution Language. *See* BPEL

D

Deployment Problems 4, 39

E

EAI 27, 35, 46

EAI Scenarios 4, 35

ebXML 36, 46

EDI 36, 46

Electronic Business XML. *See* ebXML

Electronic Data Interchange. *See* EDI

Enterprise Application Integration. *See* EAI

Enterprise Architecture 3, 5, 6, 10, 11, 42, 46

Enterprise-Capable Products 4, 30

Enterprise-Capable Systems 4, 33

Enterprise Integration 3, 5, 6, 7, 42

Enterprise Service Bus. *See* ESB

ESB 6, 27, 28, 29, 32, 46

Extensible Markup Language. *See* XML

F

FE 35, 46

FEA 10, 46

Federal Enterprise Architecture. *See* FEA

Frameworks 3, 4, 10

Further Education. *See* FE

G

General Practicioner. *See* GP

GP 46

I

Information Technology. *See* IT
Internet Protocol. *See* IP
IP 46
IT 5, 6, 9, 10, 13, 14, 15, 16, 17, 22, 23, 34, 37, 40, 41, 43, 46, 47
IT Governance 16, 43
IT/SOA Governance 4, 16

K

Key Performance Indicator. *See* KPI
KPI 25, 46

M

Massachusetts Institute of Technology. *See* MIT
Ministry of Defence Architecture Framework. *See* MODAF
MIT 5, 46
MODAF 11, 45, 46

N

National Health Service. *See* NHS
National Programme for IT. *See* NPfIT
NHS 37, 46
NHS CFH 37, 46
NHS Connecting for Health. *See* NHS CFH
NPfIT 37

O

Open Group Service Integration Maturity Model. *See* OSIMM
OSIMM 19, 45, 47, 48

P

PDA 47
Personal Digital Assistant. *See* PDA

R

Registries 4, 26
Repositories 4, 14, 26
RIA 47
Rich Internet Application. *See* RIA

S

Service Oriented Architecture. *See* SOA
SOA 3, 4, 5, 6, 11, 16, 17, 18, 19, 22, 23, 25, 27, 28, 36, 39, 42, 43, 44, 47
SOA Governance 4, 6, 16, 43

T

The Open Group Architecture Framework. *See* TOGAF

Third Part Logistics. *See* 3PL

TOGAF 11, 12, 45, 47

U

UK 35, 45, 47

United Kingdom. *See* UK

United States of America. *See* USA

USA 13, 47

V

Virtualization 41

W

Web Services 45

X

XML 19, 26, 46, 47

This work was funded by Digital 20/20. Further information is available at http://www.digital2020.org.uk.

This booklet was written and produced by Colin and Christine Smythe of Dunelm Services Ltd (http:///www.dunelm.com), at 34 Acorn Hill, Stannington, Sheffield, S6 6AW, Tel: 0114-2334009, E-mail: colin@dunelm.com.