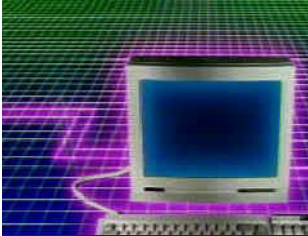


Dunelm

Services Limited



Consultants in Applied Research & Development for
Information & Communications Technology



Session 1: Concepts for SOA ***Service Oriented Architecture:*** ***Preparing Your Business &*** ***IT Products***

2009 © Dunelm Services Limited

PUBLICATION INFORMATION

The publication history for this booklet is:

<i>Version</i>	<i>Date</i>	<i>Summary of Amendments</i>
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

This series of booklets, entitled *Service Oriented Architecture (SOA): Preparing Your Business & IT Products* is aimed at two types of organisation. The first type will be about to investigate how SOA could be used to improve the range of the systems and services offered to their user base. For example, by making it easier to integrate established services and so simplify the way in which they are used. The second is product development companies who will be about to consider how SOA could be used to improve their product range. For such companies there are several ways in which the use of SOA could be used to either renovate established products or to create a new range of products. Both the business opportunities and technical considerations for SOA are addressed in this series of podcast. The overall aim is to help you, the listener, to decide if SOA is a potential business solution.

This series consists of six sessions. This is the first session and addresses the Concepts for Service Oriented Architecture. The other five sessions look at the Business Perspective and Opportunities, the SOA Maturity Model, Enterprise Integration, three Cases Studies and, finally, the Core Technology.

TABLE OF CONTENTS

Publication Information	2
Executive Summary	3
1. Introduction	5
2. What is SOA?	8
3. Why is SOA Important?	10
4. When Should SOA be Used?	13
5. Who is Adopting SOA?	15
6. Where Should SOA be Used?	17
7. How is SOA Achieved?	19
8. Gartner Technology Cycle	20
9. Business Perspective	23
10. SOA Maturity Models	25
11. Enterprise Integration	27
12. Case Studies	29
13. Core Technology	31
14. In Conclusion	34
Appendix A – Bibliography	39
Appendix B – Acronyms	40
Index	41

1. INTRODUCTION

During the last eight years there have been a large number of new technologies created that improve the way in which the Web can be exploited commercially and socially. This starts with XML data, SOAP messages and Web Services and leads into SOA. To these are added software as a service, virtualisation, grid computing, utility computing and cloud computing. At the same time the browser has become the ubiquitous user interface leading to a variety of rich Internet applications and rich mobile applications. While the pace of development of these new technologies accelerates there have been significant changes in the expectations for how an organization must operate. Legislative requirements place significant reporting and auditing demands on all organizations. The number of mergers and acquisitions increases as organizations strive to increase their market reach and maintain the pace of product innovation. IT makes all of these changes possible. IT is not the driver, but the facilitator.

Service oriented architecture is an attempt to look at how these technologies can be harnessed to improve the business processes of an organization. This gives rise to two perspectives: the user and the supplier. The user needs systems, applications and tools that improve the way they operate and makes them more competitive. The supplier wants to create products that users will buy, that are easy to maintain and which are innovative. SOA appears to be one area where users and suppliers believe

they have mutual benefit.

There are three learning objectives for this first Session. The first is to introduce service oriented architecture and explain its relationship to current best practices in terms of IT service provision and business operation. The focus is on the business benefits and opportunities and not the technology that makes SOA a reality. We look at the what, why, when, who and where of SOA. Naturally, we only touch briefly on each of these but it is sufficient to provide the context for the detailed materials covered in the later sessions.

The second learning objective is to explain how SOA can be used to drive technical innovation and support business process change. Other sessions will look in more details at the how, for example Session 6, but a quick look at the types of technology is important to give a rounded introduction to SOA. A small part of the Gartner technology hype cycle for 2008 is discussed to provide a timeline for when SOA approaches can be expected to become mainstream.

The third learning objective is to give an overall perspective of the set of materials presented in the entire set of six sessions. The first two sessions set the business context for using SOA and give guidance on the business opportunities and benefits that can be obtained by such an approach. Session three looks at SOA maturity models that can be used by an organization to identify its readiness for SOA or to help determine where and how to improve its usage of SOA. Session four looks at the broader issue of enterprise integration and sets SOA within the broader

IT context for an enterprise. Session five looks at three case studies that show how SOA can be adopted to improve business competitiveness and to support the development of innovative products. Session six is where all of the technology nitty-gritty is addressed.

2. WHAT IS SOA?

Service oriented architecture positions IT services as the primary means through which access to information and business capability is represented in support of the realization of the business operations. SOA is an approach to IT solutions that is driven by the Organization's needs. IT solutions are delivered using small modules called "services" that support the explicitly described needs of the organisation. These services are designed for use by more than one IT solution. Over time new IT solutions can be built from already existing services. Existing applications can expose themselves as a set of IT services, whilst new IT systems may be implemented in smaller modules with IT service interfaces included in the original designs.

Change is a constant in any business environment and being able to respond to change without investing considerable time and effort in modifying IT systems is a key benefit of SOA. Furthermore, SOA enables a business to not only transform internal systems to be more service oriented, but also permits collaboration amongst businesses, other partners and third-party suppliers of useful business functions in a seamless, non-disruptive fashion.

SOA is one framework (a collection of guiding design principles, a vocabulary of types of components and assembly rules) used to define and build IT systems. As the name implies, SOA is based on building systems that are composed or built from services. Like other architectural frameworks, SOA is defined in terms

of a set of specific characteristics that identify how the services are defined (the vocabulary of types) and how they interact (the assembly rules). The key characteristics of SOA are: defined, communicating, reusable, abstracting, composable, coarse grained, document oriented, loosely coupled, autonomous, stateless, discoverable, managed and providing quality of service (described in Session 2). Where there is a range of possible values for a characteristic e.g. state, etc., SOA defines specific measures of the characteristics (design principles). While these are the preferred values for these characteristics, there are no hard and fast rules to say that an approach that violates one of the characteristics or principles does not correspond to SOA.

As an architectural framework, SOA does not define the technologies used to build an operational system based on SOA. Web Services, while the most common set of technologies used to implement SOA, is not the only one possible. Equating SOA and Web Services is incorrect; SOA is the model, Web Services is a realisation of the SOA model used to build operational IT systems using a particular technology platform.

If businesses look to using SOA to support their operational activities then suppliers of IT systems, applications and tools must develop products so that they can be used in the SOA context. This can create great uncertainty for the supplier because it may mean their product set has to be completely redesigned i.e. product revolution not evolution. This has great commercial risk. Therefore, creating products that support SOA has to be carefully planned from the technical, market and commercial perspectives.

3. WHY IS SOA IMPORTANT?

Many large and medium sized organizations are interested in SOA because it promises to reduce the complexity of integrating systems while at the same time aligning the enterprise business processes with the underlying technology used to deliver those processes. However, the lack of a common blueprint for SOA is preventing many organizations from adopting it.

Many organizations must provide enterprise services to their ecosystem of many different people. Typically, the varying needs of this ecosystem, is dependent upon the management of a complex heterogeneous environment comprised of new and legacy IT systems. Each IT system may require a different server, operating system, messaging platform, database, and application. Data must be shared among these systems to support the enterprise processes, so systems integration is a necessity.

Systems integration is difficult to achieve and manage. As the number of enterprise services (and supporting IT systems) increase, the number of integrations required increase at an even faster rate. At first, traditional point-to-point integration might seem easy to manage, but over time, this approach becomes extremely complex, making it difficult for an institution to analyse the impact of replacing an enterprise service, upgrading the IT system that supports it, or opening the service up to new and different users. Institutions are often hesitant to undergo any transformation because of this underlying complexity, so

they become less responsive and less able to keep up with the demands of their user ecosystem. Furthermore, the traditional style of integration was usually undertaken in an inconsistent manner, through ‘nightly batch file updates’, remote-procedure calls, database updates, and the like. As a result, any data that is modified is rarely up-to-date in all systems, leading to errors in fees, course enrolments, addresses, and payments.

In a services-oriented approach, there is a continuum of recommended practices that can be used to describe activities related to application integration, development and deployment that yields the benefits of a flexible, loosely coupled architecture. Some of the core recommendation practices have been codified as a set of principles. There have been various definitions of these principles including the eight common principles articulated by Thomas Earl [Erl, 07]. Foremost amongst these principals is that of loose coupling. In other words, it does not matter how services communicate or what the communications looks like, it only matters that the communication is documented and well understood. For example, a website that was receiving and processing a request was considered SOA. It didn’t matter if the request was in an HTTP envelope with a message that could only be parsed in a proprietary way. Many of the early implementations of service-oriented approaches have been at this end of the spectrum.

However, service oriented architecture requires more than a services approach. In addition, important fundamental aspects of SOA must include:

- Alignment of business services with application services;
- Use of standardized mechanisms for discovering & communicating between services;
- The importance of information consolidation as an integral part of a service oriented solution;
- Exposing business capabilities as services, permitting the creation of new business processes.

SOA brings a strong focus on service to IT systems at the enterprise, but it does not do away with the need for enterprise architecture. In other words, it is still important to understand application capabilities, data capabilities and integration requirements between the various IT systems at the enterprise. SOA helps build useful services on top of these underlying IT systems, providing new ways for integrating the various underlying systems in support of a services paradigm. Presence of enterprise architecture greatly simplifies the creation of a service-oriented architecture with its resultant benefits.

4. WHEN SHOULD SOA BE USED?

Service oriented architecture is not the right answer for all business facing technology problems; however it does offer value in a variety of situations. Some of the more common situations where SOA lends value are:

- New IT solutions requiring integration of Enterprise capability from disparate IT systems and programming models. This is particularly true during acquisitions and mergers in which the ability to integrate IT infrastructures is a key part of the ‘due diligence’;
- Increased efficiency in working with business partners for driving revenue, saving costs and for off-loading non-core functions. For many successful organizations, it is the way in which they use IT that makes them successful;
- Improving employee productivity by providing information they need when and where they need it. Many organizations collect data about how they perform, far fewer analyse that data and so they cannot get the relevant information into the hands of the people who can use it best;
- Leveraging existing enterprise assets by making them accessible for reuse outside their original purpose. Some innovation opportunities are lost because many people in an organization are unaware of the assets that are owned elsewhere within the organization;

- Permitting organizations to change quickly in response to changes in the marketplace or competitive threats. The ability to change, when required, quickly and efficiently has always been a characteristic of an excellent organization;
- Maintaining cross-institutional records, improving the accuracy of management reporting and auditing, and reducing the time to produce management reports.

To contrast, some of the situations where SOA is typically not the best answer are:

- Only a small percentage of an organization's IT budget is spent on integration activities;
- A majority of an organization's processes are manual, with little opportunity for automation;
- The operation of an organization is managed by one or two Enterprise Resource Planning (ERP) applications with little or no integration requirements.

Organizations that undertake SOA projects typically start with a small project, restricted to one or two departments within the organization, and only upon achieving success and establishing a positive Return On Investment (ROI), expand to the rest of the organization. Succeeding with SOA is not always about transforming the entire organization. Frequently, focusing on a specific problem area or objective is an excellent starting point for increasing the service orientation of an organization.

5. WHO IS ADOPTING SOA?

The primary driver for the adoption of SOA is support for enterprise systems and enterprise integration. There are two perspectives on this adoption:

- The user perspective in which SOA is required to improve the operational capabilities of the enterprise as a whole. Typically, there are three types of user:-
 - Large/medium size organizations which have a myriad of systems to integrate and which have increasing demands for accurate and timely management and audit reporting
 - Small/medium sized organizations that require new business services and may want to integrate these to some in-house solutions
 - New micro organizations that require standard business services and want to have access to these on a per-usage basis thereby controlling costs and who do not want to be responsible for management and maintenance of those services;
- The supplier perspective and the need to increase market share, to enter new markets and to continue innovation. In the case of large system suppliers, such as Oracle, SAP, etc. there is an increasing awareness that they must supply simpler and cheaper solutions. They also recognise that their

generic solutions cannot be used in specialist sectors unless they have sector specific solutions. For example, SAP can provide generic logistics management functionality tailored to a specific market, inventory management, but they do not have the detailed knowledge that a specialist provider would have of that sector. SOA provides the framework for identifying the key service components. A simple generic solution can easily be replaced, once a suitable one has been found, by a more powerful specialist service. As small suppliers are seeing their traditional specialist markets under attack by the large suppliers, there is an integration opportunity. If a product is service-oriented then it is possible to integrate with large enterprise solutions with relatively modest interoperability implementation work. Therefore, small suppliers need to adopt SOA to protect their own markets and to exploit new integration opportunities.

6. WHERE SHOULD SOA BE USED?

During the past 12 months there have been many organizations claiming that they have SOA-based products. Indeed, reference to SOA appears to have been an essential part of any product pitch. This is typical of the marketing hype that accompanies a new technology. However, SOA is not just a technology. The adoption of SOA is part of a review of the way a business operates. Similarly, using SOA-based products must reflect the way an organization wants to operate: it should not dictate the way in which the business must operate. Many of the currently available SOA products are not big ‘SOA’. Instead they are small ‘soa’.

Small ‘soa’ is the starting point for migrating to big SOA. Small ‘soa’ starts as a set of service wrappers around the original applications. These wrappers will use Web Services but they typically suffer from poor interoperability due to the dependence on the implementation platforms. Big SOA allows an organization to create new business processes by reusing services with minimal new implementation work. New services from other suppliers can also be used with equal ease i.e. SOA avoids supplier lock-in. Little ‘soa’ has none of these characteristics. In fact, little ‘soa’ can be used by a supplier to re-introduce the ‘one-stop shop’ syndrome.

Proprietary use of SOA may be worse than no SOA. Sealed service interfaces that are only available to products from the same supplier is just another form of monolithic design. It may

offer more flexibility in the future but this becomes a decision of market and competition positioning.

SOA can be used in every computer-based system but this does not mean that every part of the system uses SOA. For example, real-time systems should not use SOA because of the performance degradation. However, if the real-time system is required to externally report internal logging information to the managing enterprise architecture then SOA should be used. For example, some street lighting systems operate in real-time to alter their lighting conditions to reflect the ambient conditions. Each street-light will also report to its remote management system and this information is used by a local Council to produce operational reports. This reporting could be implemented using SOA.

Suppose a piece of software is used as a stand-alone tool, applications or system. Does this need to be SOA-savvy? From a user's perspective there may be no advantage when using the product. However, an SOA-based implementation may mean that the product is easier to support remotely or could be supplied more cheaply. Both of these have indirect benefit to the user. When considering the use of SOA it is the entire product lifecycle that must be examined. Similarly, the new product development strategy should also be considered because using SOA may provide better opportunities for innovation.

7. How is SOA ACHIEVED?

At the simplest, technological, level, SOA is achieved using the Web and the Internet. Over time two alternative approaches have been developed and these now have their own advocate and adoption communities. In both cases the Extensible Markup Language, more commonly referred to as XML, is the preferred data representation format and HTTP is used as the underlying Internet protocol: remember that this is all about extending the way traditional Web servers operate. The split is into the SOAP and REST communities. The SOAP community use SOAP messages but there are a myriad of support Web Services standards to define all sorts of information exchange capabilities e.g. security, reliable messaging, exchange policy, etc.

The Representational State Transfer (REST) community abhors the use of SOAP, and thus the related specifications, claiming it is overly complex and changes the nature of true Web-based interactions. REST is defined as a collection of network architecture principles that can be bound to any collection of appropriate technologies. However, HTTP is always used. It is the SOAP approach that is usually assumed when the term Web Services is applied. RESTful approaches are collectively referred to as Web Oriented Architecture (WOA).

However, this is just the technological perspective at its simplest. SOA is really about using services to achieve business objectives. Using Web Services is not SOA; it is a part of the way in which may SOA is realised.

8. GARTNER TECHNOLOGY CYCLE

The Gartner Hype Cycle for Emerging Technologies was first published in 1995. The form of the cycle has been subtly changed over that time but it is still one of the most widely cited visualizations for predicting technology adoption. The predictive accuracy of the cycle is hotly debated. The cycle attempts to show when new technology will become mainstream taking into account that early adoption of any technology is always preceded by inflated expectations and disillusionment. One of the fascinations with the cycle is the technologies that are initially missed but which later appear with shortened timescales to adoption e.g. Web 2.0.

The July 2008 version had four pertinent revelations, as shown in Figure 1:

- Basic Web Services is now in common deployment. This is not to say that all of the problems for deployment and interoperability have been resolved. Instead, when appropriate, Web Services deployment should be seen as an expected form of service delivery. For example, the release of software updates should use the Web;
- SOA is itself now into the *Slope of Enlightenment* and should expect to enter broad adoption within the next 2-5 years. This implies that, given the time taken to re-engineer a system or product to support full SOA, strategic planning for the use SOA should be completed before the end of 2009;

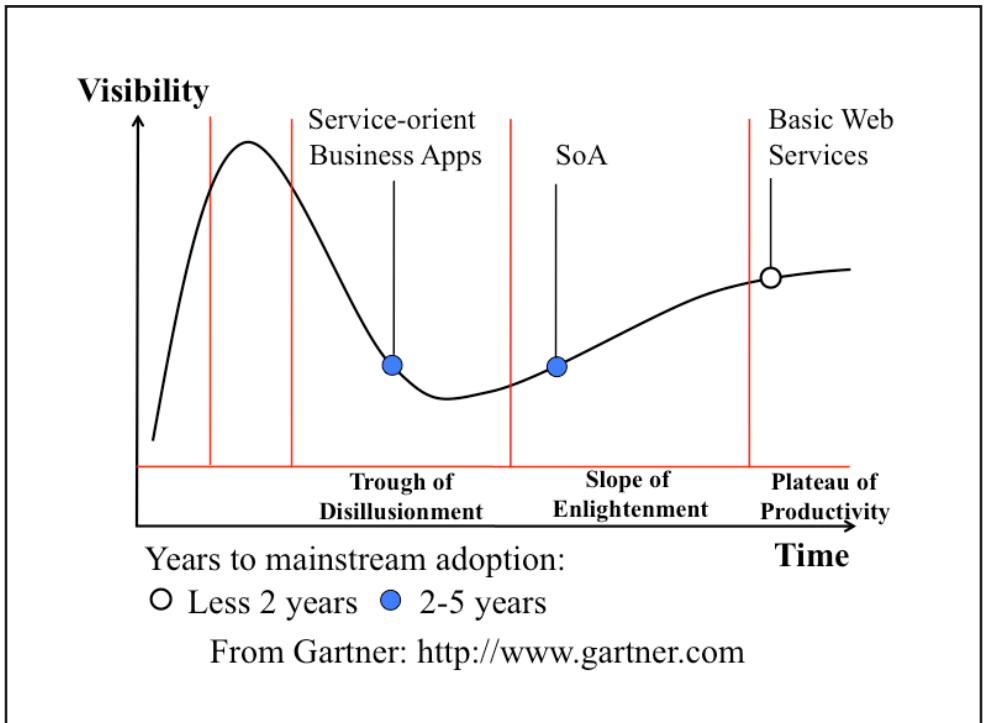


Figure 1 Partial Gartner technology ‘hype’ cycle for 2008.

- Deployment of service-oriented business applications is considered to be in the *Trough of Disillusionment* but with expected broad adoption within the next 2-5 years. Therefore, as with use of SOA, strategic planning for the development and/or deployment of SOA-based business applications should be underway;
- Cloud computing is still in the earliest phases of the cycle, i.e. *Technology Trigger*. However, it too has a 2-5 year adoption prediction. Therefore, cloud computing should form some part of the strategic planning for the use of SOA.

It would be folly to take these timescales too seriously. Instead, they should inform where the community as a whole perceives the state-of-the-art. If a company wishes to be a leading edge adopter or supplier of such organizations then these timescales may be too long. It is important to interpret these predictions in the context of your market sector and positioning within that sector. Early adopters should be using SOA/Cloud Computing now whereas late adopters will, if still operating, using the technology in the next five years.

9. BUSINESS PERSPECTIVE

In the second session on SOA the focus is on clarifying the business perspective and identifying the opportunities that could arise from using SOA. Using new technology solely for the sake of the technology itself is a recipe for organisational disaster. There must always be some clear identifiable advantage for using a new technology. This doesn't have to take the form of immediate cost savings or producing new revenue streams but it should have direct and immediate benefit. For example, it might improve management reporting thereby enabling the management team to identify ways in which to make the organization more efficient.

There are three learning objectives for Session 2, namely:

- The first is to explain the business process issues that should be considered when preparing to use systems based upon SOA. The use of SOA should improve one or more business processes. If it does not then its use of SOA is difficult to justify;
- The second is to explain that SOA may not be the answer to all business IT problems and how to identify when and where SOA can be of benefit. In many cases, SOA may only be worthwhile in part of a system. Adopting SOA can be expensive in terms of up-front investment and so careful evaluation and planning is essential to minimise the required investment;

- The third is to explain the relationship between Business Process Modelling and SOA and to identify the possible return on investment. There must be some commercial advantage to undertaking SOA otherwise the investment is wasted. More importantly, there has to be some clear return on investment that is quantifiable in terms of amount and timescale.

An important characteristic of a thriving organization is the ability to continuously evaluate and improve performance. Small regular improvements are important. SOA-savvy products should be designed to enable these improvements.

10. SOA MATURITY MODELS

One of the most significant problems with using architectures and frameworks is trying to work out how they fit into the current systems. In most cases a current system has to be migrated to the new form as opposed to creating a new system from scratch. For many years, the software industry has advocated the use of maturity models. In the case of SOA, a maturity model is needed to enable organizations to identify the extent to which they either use or could use SOA. The same model can then be used to help organisations plan how to improve their use of SOA. Apart from being able to describe the level of maturity the model also needs to provide some metrics that are used to measure the level of maturity. It is these measures that an organisation then uses to determine their current level of maturity. Those same metrics are then used to guide an organisation in how to improve their level of maturity to increase business productivity.

There are three learning objectives for Session 3, namely:

- The first is to introduce and explain the SOA Maturity Model from the Open Group. This is called the Open Group Service Integration Maturity Model, the OSIMM, and was developed with IBM [OSIMM, 06] . The OSIMM is one of several maturity models for SOA and but is not an attempt to suggest that it is the best. Instead the aim is to identify the characteristics required for such a model;

- The second is to explain how an organization can gauge where it is on the maturity model. An important feature of a maturity model is that it must be easy to map current systems onto it and also allow new system designs to be evaluated. It must be possible to map the properties of a system onto a set of points in the maturity model so that the system SOA maturity can be expressed. The next step is that the maturity model must indicate what changes have to be made to improve the SOA capabilities of a system;
- The third is to explain how an organization can use the maturity model to plan the development of its own systems and/or products and to be more innovative. A maturity model is a tool for use by the senior technical team of an organization. Two simple examples will be used to explain how the assessment methodology in the OSIMM can be used to evaluate SOA maturity.

11. ENTERPRISE INTEGRATION

In this fourth session of the SOA series 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 it is “the organizing logic for business processes and IT infrastructure reflecting the integration and standardization requirements of the firm’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.

In Session 4, the three learning objectives 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.

12. CASE STUDIES

There are many examples of systems based on SOA, in its many forms, already deployed. Also, many suppliers are looking at how their current range of products can either use SOA or be integrated with SOA-based solutions. This isn't to say that these systems make full use of SOA. However, SOA is already being adopted and whether you are a user or supplier, you need an opinion about when and how your organization will adopt SOA.

In Session 5 three case studies are worked through to show how SOA can be used. In Session 5, the learning objectives are to:

- Show how a set of service definitions is used to provide enterprise integration for Oracle's Enterprise Student Administration Integration Pack (SAIP). The SAIP uses an e-learning interoperability specification from the IMS Global Learning Consortium to define how student information, course information, etc. should be exchanged with other related systems using Web Services;
- Discuss some of the different approaches by which SOA was used to drive product innovation in a number of SMEs. In many cases, making a product SOA-savvy is revolution and not evolution. Typically the new product will need to have its architecture rethought and so this is an ideal opportunity to consider what the 'next generation' of product should also provide. One important consequence of the next generation is that established clients must be given an easy migration

from the old to the new otherwise customers will either not make the change or prefer to look at alternative solutions;

- Discuss how the Amazon Web Services can be used to extend the capabilities of a system. Amazon provide a number of different commercially available services e.g. cloud computing, e-commerce, on-demand workforce, etc. Many of these are made available through Web Services and so they can be used to realise just a part of a business process as opposed to the whole process.

For each of these Case Studies we look at the commercial benefits and outline the technical issues that have to be resolved.

13. CORE TECHNOLOGY

In the previous five sessions the significance of the implementation technology for SOA has not been addressed. However, the implementation of a system based upon SOA must use technology that can reflect the characteristics of SOA e.g. loose service coupling, etc. Poor implementation will defeat the key SOA system properties such as flexibility i.e. the ability to create new business processes by novel reuse of established services. In Session 1 the 2008 Gartner Technology Hype Cycle was described in which the use of basic Web Services was identified as established practice. In this Session the technology used in basic Web Services is described.

The core Web technologies are the minimal set needed to provide service-oriented computing. The core defines the mechanism for passing messages between a client (service consumer) and a service (service provider), and the way to describe services (the operations/messages they support). The Core consists of:

- The Extensible Markup Language (XML) with XML Schema used to describe the data to be exchanged;
- SOAP as the protocol for the exchange of messages described in XML over a distributed network e.g. the Internet;
- The Web Services Description Language (WSDL) is used to describe Web Services in terms of the message exchange patterns.

There are an almost endless number of ways in which Web Services can be implemented. Therefore, best practice guidance is essential. The Web Services Interoperability Organization (WS-I) is an open industry consortium chartered to establish Best Practices for Web Services interoperability, for selected groups of Web Services standards, across platforms, operating systems and programming languages.

Within the context of this series on SOA there are three learning objectives for this Session, namely:

- The first is to explain what implementation technologies should be adopted in the short and mid term. The short term is assumed to be 1-2 years whereas the mid-term is the next 2-5 years. Given the pace of technology development and the inherent uncertainty in predicting technology changes there is little benefit in thinking beyond five years. Indeed, for most of us, any return on investment must be achieved within the short term and so the relevance of long term technology developments are heavily coloured by short term experiences;
- The second is to explain the issues and concerns to be considered when looking to adopt Web Service solutions. When the first Web Services solutions were being deployed there were few alternative technologies to consider but the tools for these technologies were primitive. Throughout this session these issues of compatibility and interoperability will be identified and the recommended best practices to resolve them will be suggested;

- The third is to explain how technical interoperability must be addressed to support innovation. Recall that one of the objectives of SOA is to enable a plug-and-play approach for services based upon the reuse and combination of services. At a technical level interoperability is essential. Interoperability means that two systems can exchange the required information without new implementation work irrespective of which platforms are used to support the systems. This means that the systems must use the appropriate service capabilities defined in terms of both the interface (or application programming interface) and the protocol (or the Web Service). Interoperability must be addressed at many different levels. However, once interoperability for each service has been established then new business processes can be created by innovative combination of those services.

Finally, of all the material covered in the six sessions, this is the content that is most susceptible to change. There are a lot of relevant standards being developed and in many cases these are undergoing significant revision as we gain more implementation experience. Likewise, the tool-sets that make it possible to create a Web Services solution are also being changed and improved. Therefore, it is important to monitor developments closely.

14. IN CONCLUSION

Service oriented architecture is the latest attempt to look at how Web-based technologies can be harnessed to improve the business processes of an organization. This gives rise to two perspectives: the user and the supplier. The user needs systems, applications and tools that improve the way they operate and makes them more competitive. The supplier wants to create products that users will buy, that are easy to maintain and which are innovative. The what, why, when, who, where and how of SOA is summarised as:

- What is SOA? – SOA is an architectural principle that positions IT services as the primary means through which access to information and business capability is represented in support of the realization of the business operations. SOA is an approach to IT solutions that is driven by the Organization's needs. IT solutions are delivered using small modules called "services" that support the explicitly described needs of the organisation. These services are designed for use by more than one IT solution. Over time new IT solutions can be built from already existing services. Existing applications can expose themselves as a set of IT services, whilst new IT systems may be implemented in smaller modules with IT service interfaces included in the original designs;
- Why is SOA important? – many large and medium sized organizations are interested in SOA because it promises

to reduce the complexity of integrating systems while at the same time aligning the enterprise business processes with the underlying technology used to deliver those processes. However, the lack of a common blueprint for SOA is preventing many organizations from adopting it. Many organizations must provide enterprise services to their ecosystem of many different people. Typically, the varying needs of this ecosystem, is dependent upon the management of a complex heterogeneous environment comprised of new and legacy IT systems. Each IT system may require a different server, operating system, messaging platform, database, and application. Data must be shared among these systems to support the enterprise processes, so systems integration is a necessity. SOA brings a strong focus on service to IT systems at the enterprise, but it does not do away with the need for enterprise architecture. In other words, it is still important to understand application capabilities, data capabilities and integration requirements between the various IT systems at the enterprise. SOA helps build useful services on top of these underlying IT systems, providing new ways for integrating the various underlying systems in support of a services paradigm. Presence of enterprise architecture greatly simplifies the creation of a service-oriented architecture with its resultant benefits;

- When should SOA be used? – some of the more common situations where SOA lends value are for:
 - New IT solutions requiring integration of Enterprise

capability from disparate IT systems and programming models

- Increased efficiency in working with business partners for driving revenue, saving costs and for off-loading non-core functions
- Improving employee productivity by providing information they need when and where they need it
- Leveraging existing enterprise assets by making them accessible for reuse outside their original purpose
- Permitting organizations to change quickly in response to changes in the marketplace or competitive threats
- Maintaining cross-institutional records and reducing the time to produce management reports;
- Who is adopting SOA? – the primary driver for the adoption of SOA is support for enterprise systems and enterprise integration. There are two perspectives on this adoption: the user perspective in which SOA is required to improve the operational capabilities of the enterprise as a whole. The supplier perspective and the need to increase market share, to enter new markets and to continue innovation;
- Where should SOA be used? – many organizations claim that they have SOA-based products. Indeed, reference to SOA appears to be an essential part of any product pitch. This is typical of the marketing hype that accompanies a new technology. However, SOA is not just a technology. The

adoption of SOA is part of a review of the way a business operates. Similarly, using SOA-based products must reflect the way an organization wants to operate: it should not dictate the way in which the business must operate. Many of the currently available SOA products are not big 'SOA'. Instead they are small 'soa'. Small 'soa' is the starting point for migrating to big SOA. Small 'soa' starts as a set of service wrappers around the original applications. These wrappers will use Web Services but they typically suffer from poor interoperability due to the dependence on the implementation platforms. Big SOA allows an organization to create new business processes by reusing services with minimal new implementation work. New services from other suppliers can also be used with equal ease i.e. SOA avoids supplier lock-in. Little 'soa' has none of these characteristics. In fact, little 'soa' can be used by a supplier to re-introduce the 'one-stop shop' syndrome. Proprietary use of SOA may be worse than no SOA. Sealed service interfaces that are only available to products from the same supplier is just another form of monolithic design. It may offer more flexibility in the future but this becomes a decision of market and competition positioning;

- How is SOA achieved? – at the simplest, technological, level, SOA is achieved using the Web and the Internet. Two alternative approaches have been developed and these now have their own advocate and adoption communities. In both cases the Extensible Markup Language, more commonly referred to as XML, is the preferred data representation

format and HTTP is used as the underlying Internet protocol: remember that this is all about extending the way traditional Web servers operate. The two communities then split into the SOAP and REST communities. The SOAP community use SOAP messages but there are a myriad of support Web Services standards to define all sorts of information exchange capabilities e.g. security, reliable messaging, exchange policy, etc.

APPENDIX A – BIBLIOGRAPHY

- [Erl, 07] *SOA: Principles of Service Design*, Thomas Erl, Prentice Hall, 2007, ISBN -13-234482-3.
- [OSIMM, 06] *The Open Group Service Integration Maturity Model (OSIMM)*, Draft, The Open Group, 2006.

APPENDIX B – ACRONYMS

ERP	Enterprise Resource Planning
HTTP	Hypertext Transfer Protocol
IT	Information Technology
OSIMM	Open Group's Service Integration Maturity Model
REST	Representational State Transfer
ROI	Return On Investment
SAIP	Student Administration Integration Portal
SME	Small Medium Enterprise
SOA	Service Oriented Architecture
WOA	Web Oriented Architecture
WSDL	Web Services Description Language
WS-I	Web Services Interoperability Organisation
XML	Extensible Markup Language

INDEX

B

Business Process Modelling 24

C

Case Studies 4, 7, 29, 29–30

Cloud computing 5, 21, 30

Core Technology 3, 4, 31–33

E

Enterprise Architecture 27

Enterprise Integration 3, 4, 27–28

Enterprise Resource Planning. *See* ERP

ERP 14, 40

Extensible Markup Language. *See* XML

G

Gartner technology hype cycle for 2008 6

Grid computing 5

H

HTTP 11, 19, 38, 40

Hypertext Transfer Protocol. *See* HTTP

I

Information Technology. *See* IT

IT 1, 3, 5–10, 12–14, 23, 27, 34–36, 40

M

Maturity Model 3, 25–26, 39, 40

O

Open Group's Service Integration Maturity Model. *See* OSIMM

OSIMM 25, 26, 39, 40

R

Representational State Transfer. *See* REST

REST 19, 38, 40

Return On Investment. *See* ROI

ROI 14, 40

S

SAIP 29, 40

Service Oriented Architecture. *See* SOA

Small Medium Enterprise. *See* SME

SME 40

SOA 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36, 37, 39, 40

SOAP 5, 19, 31, 38

software as a service 5

Student Administration Integration Portal. *See* SAIP

Systems integration 10

U

Utility computing 5

V

Virtualisation 5

W

Web Oriented Architecture. *See* WOA

Web Services 5, 9, 17, 19, 20, 29, 30, 31,
32, 33, 37, 38, 40

Web Services Description Language.

See WSDL

Web Services Interoperability Organisation.

See WS-I

WOA 19, 40

WSDL 31, 40

WS-I 32, 40

X

XML 5, 19, 31, 37, 40

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.