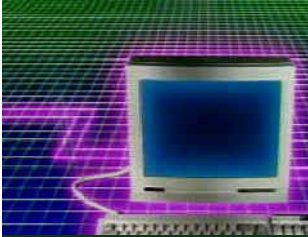


Dunelm

Services Limited



Consultants in Applied Research & Development for
Information & Communications Technology



Session 2: The Business Perspective & Opportunities *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

In this second session on Service Oriented Architecture (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 organisation more efficient.

SOA is not just a technology. Apart from the typical marketing hype that follows any new terminology, SOA can also be used as a lever for understanding the business processes for an organisation. In fact, the first step in adopting SOA is to undertake an evaluation of the current business processes. From this it becomes possible to identify the ways in which the processes can be improved: some by using SOA technology.

TABLE OF CONTENTS

Publication Information	2
Executive Summary	3
1. Introduction	5
2. Commercial Imperatives.....	7
3. Enterprise Integration	10
4. “Horses for Courses”	13
5. “Divide & Conquer”	14
6. A Design Paradigm	17
7. Interoperability	20
8. Open Solutions	22
9. Product Migration.....	24
10. Business Process Modelling	26
11. Change Management	29
12. Return On Investment.....	31
13. Technology Considerations	34
14. In Conclusion.....	38
Appendix A – Acronyms	41
Index	42

1. INTRODUCTION

In many organisations Information Technology (IT) is still perceived as an endless cost centre with senior IT managers failing to explain clearly the return on investment from the IT provision. Most organisations make extensive use of IT and would struggle to survive commercially if their IT infrastructure was disabled or removed. One key consequence of an organisation considering the adoption of SOA is that the relationship between the IT provision and the business processes that it supports must be explicitly identified. The development of an SOA adoption strategy requires the organisation to understand all of its business processes, how these processes are performed and to identify those that need to improved (perhaps by making better use of IT services).

Dependence on IT systems will increase. This will create an increasing range of opportunities for new products and services. It is important for all IT product developers to investigate how their products can exploit SOA. In Session 2, we focus on the business perspective and opportunities that arise from SOA. The three learning objectives for Session 2 are:

- 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 the 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.

The material covered in this session will start by describing the commercial imperatives that organizations currently face and look at how enterprise integration is of increasing importance. SOA is not the solution for every enterprise and the creation of a services-based solution will be discussed. Service interoperability is addressed along with the significance of open solutions. The discussion then focuses on the business process implications including change management and identification of the return on investment. Finally, an initial review of the technology implications will be presented.

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.

2. COMMERCIAL IMPERATIVES

Commercial imperatives are a powerful driver for an organisation. Adoption of SOA should support those imperatives. For a product developer of computer-based systems long-term sustainability requires that

- Product development cycle times and costs must be continually reduced. Many organisations now release new versions and revisions of their products on a monthly basis. This is only possible if Web-based distribution and automated updating are used. Customers are no longer prepared to wait 6-12 months for the classic product revision period. Also, the increased complexity of computer systems, applications, tools, etc. makes them more susceptible to bugs and fixes these need to be corrected and distributed as soon as possible;
- New markets have to be identified and served. Irrespective of how large the organisation, new markets are required to sustain even small growth. The size of market is almost irrelevant provided that it can be supported profitably. This means that different revenue models are required for these new markets and so the functionality of the product must be simple to configure to fit those markets;
- Support for enterprise integration is provided. Many organisations need their range of computer systems, application, tools, etc. to be able to exchange the relevant information. Effective information flow around an

organisation is essential and so every product should be able to at least report the relevant management information to other systems.

From the user's perspective, new systems must allow the organisation to:

- Easily change its business processes. Over time every organisation changes and so its business processes must be changed appropriately. Flexibility is essential and this includes the ability to be able to tailor any system to meet the individual needs of each and every user;
- Collect, collate, analyse and report information using enterprise integration. Increased legal and financial reporting requirements for all but the smallest of organisations have been introduced in the past few years. Many sectors also have special auditing requirements e.g. food, aerospace, etc. that require long-term archiving of all operational activity associated with the products produced by the organisation;
- Control and reduce operational and maintenance costs. Many organisations depreciate their entire computer infrastructure over a period of 3-5 years. While this may be sound financial practice, for many organisations this is an unsustainable operational principle. In several markets there are systems being used that were originally installed over thirty years ago. New systems have to be designed with the same level of longevity but with the flexibility expected of modern Web-based approaches.

The use of SOA is one way in which many of the above requirements can be supplied. It is not the only answer and it is not every answer to the problems encountered by an organization. The trick is to detect when, where, and how it can be useful. What and how can then be planned and deployed.

3. ENTERPRISE INTEGRATION

For many years, Enterprise Architecture has been an important part of designing enterprise-scale systems. Enterprise Integration is one facet of Enterprise Architecture that addresses the process by which the component systems are integrated to create the enterprise solution. In session 4 we will look in more detail at Enterprise Integration and in particular the various activities undertaken. In this session we identify the business aspects that are driving Enterprise Integration. There are four clear drivers of Enterprise Integration:

- Acquisitions and mergers – these have always been an important part of corporate activity and indeed some organisations are geared to solely expand through such activities. In a modern acquisition and merger, compatibility of IT systems is one of the key areas of due diligence. Incompatible IT systems make effective business process integration difficult and expensive and so this has to be taken into account when considering the acquisition and merger;
- Audit demands for corporate governance and compliance – during the past decade there have been a number of monopoly, financial, health and safety, homeland security, etc. crises that have resulted in extensive legislation being introduced to require increasing amounts of audit and corporate governance information. This is compounded for international organisations that, by definition, trade in multiple markets in many parts of the World and as such

are subject to many legislative frameworks. This requires sophisticated audit and management reporting systems that can reach all parts of organisation and that provide accurate and timely information;

- Increase market competition and the need to identify new markets – all organisations are subjected to competition and survival is determined by the degree of success in that competition. Regular innovation gives an organisation an important advantage and helps create new opportunities in established and new markets. While it may be perverse to suggest that IT systems promote innovation, at the very least, they should not create an environment that is adverse to innovation. IT is there to serve the organisation;
- Requirements to increase efficiency and effectiveness – this takes the form of reducing operational costs, reducing time to market for services and products and generally doing more with fewer resources (whether that be equipment, people, materials, etc). Information is essential to achieving efficiency gains. However, it is certain types of information that is required. Performance metrics need to be identified, measured on a regular basis and analysed to detect deviations. Corresponding actions must then be planned and implemented and the whole process repeated. It is this continual monitoring and improvement that is a characteristic the best of breed organisations.

SOA is designed to address enterprise integration. Key features of SOA-savvy systems are that they significantly simplify the

creation and support of new business processes and make it easy to provide management information at anytime and anywhere within the enterprise.

4. “HORSES FOR COURSES”

As part of the introduction to SOA in Session 1, we discussed the “when” SOA could be adopted. SOA is not the solution for every technology-oriented business process development issue. Let’s take a closer look at the sort of situations in which SOA will not be suitable.

As we’ve discussed previously, enterprise integration is one of the strengths of SOA. Conversely, if such integration is not necessary then is SOA required? An organisation that invests only a small percentage of its IT budget on integration has little to gain by turning its internal business processes into services. A small integration budget implies that the organization is small or has very few computerised processes. In either case, the return on investment for the use of SOA would be hard to identify.

Similarly, if the majority of an organisation’s processes are manual there may be little opportunity for automation. Of course, an audit of those processes would identify which should become computerised. Also, if the operations of an organization are managed by an Enterprise Resource Planning (ERP) application there may be little or no opportunity for integration. Both of these situations are typical of smaller organisations or larger ones which are either inefficient or have strongly centralised processes.

5. “DIVIDE & CONQUER”

When designing services the standard software engineering design guidance is applied. An approach of ‘divide and conquer’ is used in which the business process is divided into its constituent services (this may be one or more services). This decomposition should reflect what services are already available so that they may be reused. If reuse is to be possible then the services should have high cohesion and low coupling. This follows the classic mantra used in software engineering for the design of subroutine and functions (it is also at the heart of object orientation).

High cohesion means that the service is functionally very focused. Also, that the set of operations have a common data set. Ideally, the service will perform a single function. For example, it may provide the ability to create, read, update and delete data about people. Another may provide similar operations on specific widgets in a warehouse database. High cohesion enables service reuse by avoiding functional duplication.

Low coupling means that a change in one service has no impact on the operation of another service. If there is an interaction, then this is only permitted through the operations of the two services. Low coupling allows the internal implementation of a service to be changed with minimal impact on the implementations of other services.

When looking at the original business process, data exchange is

a reflection of the business transactions. These transactions are how information enters and leaves the business process (this is true for manual and automated systems). If a single service is to be created to perform a business process then the operations for the service reflect the business transactions. New capabilities for a business process can be created by adding new transactions that, in turn, will require new service operations (this is always a backwards-compatible approach because the operations themselves have high cohesion and low coupling). If a business process is to be implemented by more than one service then some top-level service must be responsible for co-ordinating the constituent services.

Whenever a new service is created, it must undergo testing as a individual service but the system as a whole should be retested to confirm that there are no unexpected side effects. Creation of new services is considerably more expensive than reusing old ones. This gives rise to two categories of service: the common services and the business specific services. The former would include services such as authorisation, authentication, workflow, etc. whereas the latter reflect the business processes for the host organisation. A business specific service will use the common services but the business service itself may be reused by other business specific services.

The success of the divide and conquer approach requires an organisation to have a clear strategy for the deployment of services and business processes. If an ad-hoc development process is undertaken, as opposed to a planned approach, then

reuse will only rarely be possible and there will be no consistent architectural framework in which the services can be deployed. This will lead to poor support for the business processes that will undoubtedly result in business failure.

6. A DESIGN PARADIGM

As stressed many time before, SOA is not just a technology. It provides a set of tools to guide how a solution should be created. As a design paradigm, SOA consists of a:

- **Defined Contract** – services are defined in terms of what they do e.g. the process(es) they perform, the interface used to communicate with the services e.g. how to invoke the service to perform the process, the data that is passed to, or returned from, the service, and how the service is managed (both from a technical/implementation perspective and from a business perspective). These definitions are usually specified in some formal way, possibly a machine-processable specification and form the service contract;
- **Low Coupling** – while services need to interact to solve business problems, a service may be defined to be independent of any other service and can function without the knowledge of how other services work (except for the bindings to deployments), or it may have explicit knowledge about not only what other services are available, but how they perform their operations i.e. a service depends upon more information than just the description e.g. it may need to know about internal details. Services that are not dependent on other services are loosely coupled and this promotes ease of reuse;

- **Abstraction** – services (by their definitions) define only what process the service provides and how to communicate with it. How the service is implemented to provide the defined functionality is not defined. The service provides an abstraction boundary, hiding implementation details, or hiding the structure of the resource that the service manages;
- **Reusability** – by providing defined, discrete functionality, independent of how it is used, a service can be used (or reused) at any point in the overall business process where the functionality is needed;
- **Autonomy** – what a service requires to perform its operations is local knowledge only to the service. Autonomy is closely related to both coupling (services knowing about how other services work) and abstraction (services hiding their behaviours from their exposed interfaces);
- **Statelessness** – to fulfil a request, the operational service may need to know about historic use and invocation of the service i.e. to access data about the “state” of the business operation maintained by the service itself. The service may require no information about state i.e. stateless, or it may require knowledge of prior operations i.e. stateful. Statelessness is preferred because it requires the service interactions to always provide context and improve reusability;
- **Discoverability** – service definitions may be made available so that existing services can be found, enabling reuse and composition. Discoverability is independent of the service

itself, but part of the overall environment in which the service is defined and managed;

- Composability – a service may be combined with other services to implement the overall business process e.g. what is behind the abstraction boundary of a single service may be a composition of other services.

These features start to show the differences between little ‘soa’ and large SOA. Little ‘soa’ provides the appearance of service-based provision but it has few if any of the key characteristics to make it possible to build flexible SOA systems.

7. INTEROPERABILITY

In SOA, services are used to define what information is available and how that information can be exchanged. Services exchange information using a set of service operations. It is important that the nature of the service operations is platform independent i.e. the service can be realised on any platform. This is achieved by focusing on service interoperability and by defining what the service provides and not how it is realised. Service oriented computing is another form of system design through interface specification. A focus on interoperability means that:

- System integration requires no new implementation work for the established services – integration with other services is through their service interfaces. The service providers are implemented and so no further work on them is required. The new implementation is on the new set of service consumers;
- The system can be heterogeneous – the interface to a service defines how the service consumer and service provider interact. If the interaction occurs via a network, for example the Web, then the system is loosely coupled. This loose coupling means that the service consumers and service providers can be implemented using different platforms, implementation frameworks etc. The system can be heterogeneous provided they all implement the service interface consistently;
- The capability of the service is defined by its interoperability

interface – therefore the interface has to be sufficiently rich to provide the required service capabilities. The only requirement placed upon an implementation is that it must uphold the service contract. Therefore, the service contract must include a description of the behaviour of the system over time and the address in particular the issue of data persistence. This description of behaviour addresses the consequences of sequences of operation on the service.

Many market sectors have established not-for-profit funded consortia that focus on creating interoperability standards for key services in that market. These consortia are membership based and are used to create an impartial forum through which technical agreement on these standards can be reached. They provide an opportunity for suppliers and users to agree on key interfaces so that the sector as a whole benefits. This does not reduce competition. Instead it promotes integration of products from different suppliers. If such standards exist for your market then it is essential that you adopt them otherwise your products will be severely disadvantaged.

8. OPEN SOLUTIONS

The phrase ‘Open Solution’ is used to differentiate from a ‘Proprietary Solution’. The latter is a solution that is restricted to the owner and creator of the solution. An Open Solution is one that is made available, generally without charge, through an appropriate license. The use of a license does not change ownership i.e. the owner of the intellectual property covered in the solution. Open source software is one form of open solution i.e. distribution of the source code.

Services interfaces and the associated service interoperability definitions are ideal candidates for open solution distribution. If a service has been defined using an application programming interface and/or web services description language file (for Web Services) then these can be released using an appropriate open source license. The advantages and disadvantages of such a release are:

- The primary advantage is that publication of the interface can be used to encourage third party developers to create products that can be integrated to create a more complete enterprise solution. Large third party systems vendors can then use your products to provide specialist capability and in return you receive a new channel to market;
- A well-designed interface could become the de-facto standard for that type of service. This endows the originating organisation with substantial market credibility and can be used as an important marketing tool;

- The primary disadvantage, one that is normally cited by many market leaders, is that publication will result in loss of market share. This is particularly relevant for suppliers of one-stop solutions where significant margins are gained from the add-ons that are integrated through proprietary interfaces. If third party suppliers could provide alternatives then this will result in loss of market. Experience has shown that one-stop provision severely disadvantages the customer due to high system cost and overly expensive and restricted maintenance. It is always to the advantage of the customer to insist on the use of open interfaces.

The license under which an open solution is released is particularly important. Many organizations are using the standard licenses provided by Creative Commons (for example, this material is released under one form of Creative Commons license). There are a number of alternative open source licenses with the most extreme being the General Public License in which, for some forms, the use of such material also requires that the software using the material are made available under the same license.

For commercial uses, the most flexible open source license is one in which citation for use is required, distribution is permitted, commercial use is permitted without charge but changes to the original are not permitted. Also, there should be no release of ownership.

9. PRODUCT MIGRATION

Customers are more demanding than ever and quality of service support is an essential feature of any supplier evaluation.

Investment in IT systems/applications/tools is not just about the functionality of the product and whether or not it is state-of-the-art. Customers expect suppliers to have a clear development path for their products, a commitment to maintain older products and to provide ease of migration to any equivalent new product.

If a completely new product is under development then using SOA creates no increased problems over and above those normally encountered with any new development. However, the situation is very different if the new product is in fact an upgrade from an older product that has an established client base. In many cases the small ‘soa’ version of a product can be deployed by creating a service wrapper around the original product; Web Services would be used as the implementation technology. This is, at best, a ‘stop gap’ solution and one that many suppliers do as part of the initial evaluation process for SOA. If a product has not been designed with service-based delivery as a requirement, then converting to such an approach will usually require a total redesign. For many products, SOA is revolution and not evolution.

A number of aspects must be addressed as part of the product migration, including:

- Identify the service level expectations of the established

clients and their degree of commitment to migrating to a next generation of the product;

- Identify and define the key service support interfaces (internal and external facing) that will be used to make migration and subsequent user-support and maintenance easier to achieve;
- Determine the implications for the user-support and maintenance teams when having to simultaneously support two or more generations of product;
- Establish a Web-based upgrade policy so that new versions can be distributed easily, regularly and cost effectively. The upgrade policy should be based upon regular releases so that minor functional upgrades and bug fixes are readily available.

10. BUSINESS PROCESS MODELLING

Business process modelling plays an important role in the business process management discipline. Since both business process modelling and business process management share the same abbreviation (BPM), these activities are sometimes confused with each other.

Business process modelling addresses the process aspects of business architecture, leading to the encompassing enterprise architecture. The relationships between business processes in the context of the rest of the enterprise systems e.g. data architecture, organisational structure, strategies, etc. create greater capabilities when analysing and planning enterprise changes. For example, during a corporate merger it is important to understand the processes of both companies in detail so that management can correctly and efficiently identify and eliminate redundancies in operations.

The graphical representation of business process information has proven effective for presenting it to business stakeholders, including business analysts and system developers. Visual modelling languages used to represent business processes include Business Process Modelling Notation (BPMN) and the unified modelling language.

Business process modelling has always been a key aspect of business process re-engineering and continuous improvement approaches, such as Six Sigma. For routine business activities,

business process modelling tools from companies such as Casewise, TIBCO, etc. are used in order to represent a business process, to run a simulation of the process and for communication purposes.

A business model is a framework for creating economic, social, and/or other forms of value. The term business model is used for a broad range of informal and formal descriptions to represent core aspects of a business, including purpose, offerings, strategies, infrastructure, organisational structures, trading practices, and operational processes and policies. In the most basic sense, a business model is the method of doing business by which a company can sustain itself. That is, generate revenue. The business model is used to identify how an organization can make money by specifying where it is positioned in the value chain.

For SOA, this could be addressed as one technology framework used to realize the business processes. Business modelling tools should enable the analysis of the implementation technology, such as Web Services. These tools should allow the senior technology team to confirm that the proposed technical solution will meet the required business metrics e.g. response times, application interoperability, service choreography, etc.

Business process modelling is essential for typical enterprise deployments. By definition, enterprise systems are large and complex. Their capabilities and behaviour can only be analysed by using business process modelling tools. More importantly, enterprise systems are dynamic and so the models have to

be continually revised to match the real systems they model. Unfortunately, the modelling tools are themselves complex and so specialist support is normally required to create the initial models. Such support is expensive but there is considerable saving in the time and cost that would otherwise be wasted.

11. CHANGE MANAGEMENT

Once the business process changes have been identified, the new processes need to be created and deployed. This leads to change management: probably the most difficult part of altering a process because it is focused on getting people to change the way they work. Experience has taught us that the best way to succeed with change management is to have the people responsible for undertaking the changes involved in designing and planning the new process. Real involvement will eliminate many of the natural negative reactions to suggested change. For example, the end users must be involved with defining the requirement specification for any new system.

The biggest area of concern usually occurs when manual processes are replaced by computer-based equivalents. This leads to concerns over loss of job and significant changes in role (involving either more or less demanding work). Once a new system has been created it has to be deployed and tested. This creates another area of tension. It is not uncommon for the new and old systems to be run in parallel. This ensures that if the new system fails then key processes are still active. The disadvantage is that the workload is increased and there is the need to ensure that the process duplication does not affect other systems and processes. While SOA does not address change management, such adoption usually results in changes to how people work.

At a technical level it is essential that the development process

is itself changed. If SOA is to be used then a service-oriented design process must be developed. Change in the development process requires change management. If the development process is to be changed then it is worthwhile considering agile techniques that encourage more rapid release cycles for products. These can be amalgamated with the more traditional software development processes. Finally, it is essential that the development team work with their end users and involve them at the start of any new system development.

12. RETURN ON INVESTMENT

The senior management team for any organization will require a clear statement on the return on investment, or ROI, before any significant investment on IT is undertaken. The ROI must be expressed in business benefits and quantification is essential. This is not necessarily about saving money but the ROI must clearly identify the benefits of any increased recurrent expenditure. For SOA, the generic areas where there should be some impact on ROI are:

- Reduction in business process duplication and rationalisation of information storage. Many organisations have duplicate activity and a lot of redundant information storage. Surveys have shown that nearly 80% of archived information is never required. Removal of duplicated archived data and deep archiving can create significant cost savings. SOA should result in single point of storage of any and every piece of information because it can still be made available anywhere required within the enterprise. Duplicate business processes can now be identified and removed saving time and costs. Therefore cost savings can be found in people, materials and storage resources;
- Reduction in maintenance and upgrade costs. For large organisations, maintenance of the desktop is a significant cost. All desktop installed third party SOA systems should support Web-based maintenance upgrades. In many enterprise systems most desktop tools and applications

will be server based and so upgrading is again simple with the new versions propagating when the users log in. Maintenance and management of the desktop and server systems is addressed when designing SOA deployment. The ROI is set by lowering costs for maintenance plus improved efficiency of staff due to always having an optimally configured desktop;

- Reduction in the costs and time of producing management reports and audit trails. The adoption of SOA starts by evaluating an organisation's established business processes. For many organisations, management reporting and audit trails are responsible for a lot of the complexity of any manual activity with the need to be accurate and provide timely information. Computerisation of a process immediately resolves the collection and accuracy of such information. However, communication of this information to the relevant system in an enterprise is essential and again SOA provides this capability. One disadvantage of computerisation is information overload and so automated collation and analysis is also important. Once such automated systems are in place, processes such as just-in-time delivery become possible. The primary ROI is based upon providing more accurate management reporting such that problems and process bottlenecks can be detected and resolved more quickly;
- New revenue opportunities may be possible through new services and product features. Innovative combination of

established services and modest new development may provide a new range of services and products. Deployment of service-based products through software as a service and managed service provision opportunities should always be investigated.

Predicted ROI should always be compared with measured ROI once the new deployments have been made. Over time, the original metrics for ROI may change and so the basis of any ROI calculation must be reaffirmed regularly.

13. TECHNOLOGY CONSIDERATIONS

As the Garner technology hype cycle indicated, it is still early days for the adoption of SOA. SOA is based upon several key technologies:

- **Web Services** – a Web Service is defined by the W3C as “a software system designed to support interoperable machine-to-machine interaction over a network. Web services are frequently just Internet application programming interfaces that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services. Other approaches with similar capability to Web Services are the Common Object Request Broker Architecture, Microsoft’s Distributed Component Object Model and SUN’s Java/Remote Method Invocation. The W3C definition encompasses many different systems, but in common usage the term refers to clients and servers that communicate through HTTP and the Web;
- **Browser-based User Interface** – user interfaces rendered through a Web browser. The latest browser based user interface technologies (Microsoft Silverlight, Adobe Flex/AIR, etc.) make it possible to create Rich Internet Applications (RIAs) that start to rival the user interfaces available with desk-top equivalent applications. RIA-based applications are easier to deploy and minimise the classic problems of ensuring the desktop has all of the appropriate utilities, plug-ins, etc;

- **Enterprise Service Bus (ESB)** – ESB provides a new way to build and deploy enterprise service-oriented architectures. ESB provides an effective approach to solving common problems such as service orchestration, application data synchronisation and business activity monitoring. ESBs are the next step for middleware infrastructure technology and are attractive for enterprise solutions because they combine features from previous technologies with new services, such as message validation, transformation, content-based routing, security, load balancing, etc. The architecture of an ESB is centred on a bus. The bus provides message delivery services, based on standards such as SOAP, HTTP and the Java Messaging Service, and is typically designed for high-throughput, guaranteed message delivery to a variety of service producers and consumers;
- **Cloud Computing** – this is the provision of dynamically scalable and virtualised services through the Web. Cloud computing is a combination of Software-, Platform- and Infrastructure-as-a-Service. Cloud computing is distinct from grid computing (remote scalable computing resources), utility computing (computing functionality provided as metered resources) or autonomic computing (self managing computer systems) but is often based upon such systems;
- **Virtualisation** – this is a broad term that refers to the abstraction of computer resources. One particular form is that of the virtual server in which a complete implementation of a server and its resources and applications are realised in

software. A new server is then created by simply instantiating the virtual server again. A virtual server could be supported by several real servers or many virtual servers could be supported by a single physical server;

- **Software as a Service (SAAS)** – SAAS is a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SAAS software vendors may host the application on their own web servers or download the application to the consumer device, disabling it after use or after the on-demand contract expires. The on-demand function may be handled internally to share licenses within a firm or by a third-party application service provider sharing licenses between organizations;
- **Platform as a Service (PAAS)** – PAAS is the delivery of a computing platform and solution stack as a service. It facilitates deployment of applications without the cost and complexity of buying and managing the underlying hardware and software layers, providing all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely available from the Internet – with no software downloads or installation for developers, IT managers or end-users. PAAS offerings include workflow facilities for application design, application development, testing, deployment and hosting as well as application services such as team collaboration, web service integration and marshalling, database integration, security, scalability, storage, persistence, state management,

application versioning, application instrumentation and developer community facilitation. These services are provisioned as an integrated solution over the web;

- Infrastructure as a Service (IAAS) – IAAS is the delivery of computer infrastructure (typically a virtualised platform environment) as a service. These virtual infrastructure stacks are an example of the “everything as a service” trend and shares many of the common characteristics. Rather than purchasing servers, software, data centre space or network equipment, clients instead buy those resources as a fully outsourced service. The service is typically billed on a utility computing basis and amount of resources consumed (and therefore the cost) will typically reflect the level of activity. It is an evolution of web hosting and virtual private server offerings.

14. IN CONCLUSION

IT is still perceived as an expensive cost centre with endless capital and recurrent demands. Most organisations make extensive use of IT and would struggle to survive commercially if their IT infrastructure was disabled or removed. One consequence of an organisation considering Service Oriented Architecture (SOA) is that the relationship between the IT provision and the business processes that it supports must be expressed in benefits to the business. The development of an SOA adoption strategy requires the organization to understand all of its business processes, how these processes are performed and to identify those that need to improved (perhaps by making better use of IT services).

Commercial imperatives are a powerful driver for an organisation. Adoption of SOA should support those imperatives. For a product developer of computer-based systems long-term sustainability requires that: product development cycle times and costs must be continually reduced; new markets have to be identified and served; and support for enterprise integration is provided. From the user's perspective, new systems must allow the organisation to: easily change its business processes; collect, collate, analyse and report information using enterprise integration; and control and reduce operational and maintenance costs.

For many years, enterprise architecture has been an important part of designing enterprise-scale systems. Enterprise

integration is one facet of enterprise architecture that addresses the process by which the component systems are integrated to create the enterprise solution. SOA is designed to address enterprise integration. Key features of SOA-savvy systems are that they significantly simplify the creation and support of new business processes and make it easy to provide management information at anytime and anywhere within the enterprise.

Business process modelling is essential for typical enterprise deployments. By definition, enterprise systems are large and complex. Their capabilities and behaviour can only be analysed by using business process modelling tools. More importantly, enterprise systems are dynamic and so the models have to be continually revised to match the real systems they model. Unfortunately, the modelling tools are themselves complex and so specialist support is normally required to create the initial models. Such support is expensive but there is considerable saving in the time and cost that would otherwise be wasted.

Once the business process changes have been identified, the new processes need to be created and deployed. This leads to change management: probably the most difficult part of changing a process because it is focused on getting people to change the way they work. Experience has taught us that the best way to succeed with change management is to have the people responsible for undertaking the changes involved in designing and planning the new process. Real involvement will eliminate many of the natural negative reactions to suggested change. For example, the end users must be involved with defining the

requirement specification for any new system.

The senior management team for any organization will require a clear statement on the Return On Investment (ROI) before any significant investment on IT is undertaken. The ROI must be expressed in business benefits. Quantification is essential. This is not necessarily about saving money but the benefits of any increased recurrent expenditure must be clearly identified. For SOA, the generic areas where there should be an impact on ROI are:

- Reduction in business process duplication and rationalisation of information storage;
- Reduction in maintenance and upgrade costs; there will be a reduction in the costs and time of producing management reports and audit trails;
- New revenue opportunities may be possible through new services and product features.

APPENDIX A – ACRONYMS

BPM	Business Process Modelling
BPMN	Business Process Modelling Notation
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
GPL	General Public License
HTTP	Hypertext Transfer Protocol
IAAS	Infrastructure As A Service
IT	Information Technology
PAAS	Platform As A Service
RIA	Rich Internet Application
ROI	Return On Investment
SAAS	Software As A Service
SOA	Service Oriented Architecture
W3C	World Wide Web Consortium

INDEX

A

Acquisitions 10

Audit 10

Autonomy 18

B

BPM 26, 41

BPMN 26, 41

Business process modelling 6,
26–28, 39. *See* BPM

Business Process Modelling Notation. *See* BPMN

C

Change Management 29–30

Cloud Computing 35

Cohesion 14

Commercial Imperatives 7–9

Compliance 10

Composability 19

Coupling 14

Creative Commons 23

D

Defined Contract 17

Discoverability 18

E

Enterprise Architecture 10

Enterprise Integration 10–12

Enterprise Resource Planning.
See ERP

Enterprise Service Bus. *See* ESB

ERP 13, 26, 41

ESB 35, 41

G

General Public License 23, 41.
See GPL

GPL 41

H

Heterogeneous 20

HTTP 34, 35, 41

Hypertext Transfer Protocol.
See HTTP

I

IAAS 37, 41

Information Technology. *See* IT

Infrastructure As A Service.

See IAAS

Interoperability 20–21

IT 5, 6, 10, 11, 13, 24, 31, 36, 38,
40, 41

M

Mergers 10

O

Open Solutions 22–23

P

PAAS 36, 41

Platform As A Service. *See* PAAS

Product Migration 24–25

R

Return On Investment 31–33.

See ROI

Reusability 18

RIA 34, 41

Rich Internet Application. *See* RIA

ROI 31, 32, 33, 40, 41

S

SAAS 36, 41

Service Oriented Architecture.

See SOA

Six Sigma 26

SOA 3, 5, 6, 7, 9, 11, 13, 17, 19, 20,
24, 27, 29, 30, 31, 32, 34, 38,
39, 40, 41

Software As A Service. *See* SAAS

Statelessness 18

System integration 20

U

User Interface 34

V

Virtualisation 35

W

W3C 34, 41

Web Services 22, 24, 27, 34

World Wide Web Consortium.

See W3C

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.