mwd

macehiterward-dutton

Real SOA: critical success factors

Neil Macehiter, Neil Ward-Dutton

June 2005

The vast majority of current "service oriented thinking" concerns the evolution of application architecture – that is the focus of SOA as most people consider it today. But in order to really help align IT outcomes with business initiatives and priorities, organisations have to consider the whole of the IT organisation from a service-oriented perspective: all IT needs to be seen as service provision. To add real business value, therefore, SOA has to be considered not only from the perspective of software development. Changes to application architecture have to be supported by changes in management thinking, and organisations should evaluate new SOA technologies and tools in the context of these wider change issues.

An independent report supplied by **Systinet Corporation**

Macehiter Ward-Dutton is a specialist IT advisory firm which focuses exclusively on issues concerning **IT-business alignment**. We use our significant industry experience, acknowledged expertise, and a flexible approach to advise businesses on IT architecture, integration, management, organisation and culture.

www.mwdadvisors.com

What is service-oriented architecture (SOA), and why is it important?

SOA has generated huge amounts of interest and commentary over the past couple of years. This has culminated in wide-spread availability of a range of resources, from industry analyst reports to supplier white papers, most of which rehash the same information concerning the workings of SOAP, WSDL, UDDI, WS-*, ESB etc. This paper assumes that the reader is already familiar with the technology underpinnings of Web Services technology and associated infrastructure. The goal of this paper is to start from a business perspective, and then explain how a SOA initiative should be tackled if it is going to add real business value.

Defining SOA with a business perspective

We define SOA as follows:

SOA is the disciplined approach through which an IT organisation manages the lifecycle of IT services, and assures their delivery, in a way that reflects business process priorities.

In other words SOA explicitly recognises that all IT organisations are service providers, with customers which have a variety of business-driven IT needs: and helps them act in a systematic way which improves the overall quality of the service that they provide.

The vast majority of current "service oriented thinking" concerns the evolution of application architecture. But to add real business value, SOA cannot be just about application architecture: it must also enable a planned and systematic approach to delivering service-oriented IT.

SOA is more than Web Services…

After a considerable period, thankfully, more and more industry influencers and commentators are recognising that the basic ideas encompassed in service-oriented architecture (SOA), as it applies to application architecture, are not at all new. These ideas have roots stretching back at least thirty years. David Parnas' seminal 1972 paper "*On the Criteria to Be Used in Decomposing Systems Into Modules*"; object-oriented programming languages Eiffel, C++, Java and Smalltalk; the Distributed Computing Environment (DCE) standard; Microsoft's COM and the ill-fated OpenDoc initiative; Microsoft's DCOM and the OMG's Common Object Request Broker Architecture (CORBA standard); distributed computing middleware platforms like Tuxedo, Encina and TopEnd; all these refer to, or rely on, key ideas that are commonly expressed in today's notion of SOA.

…but SOA and Web Services is where the excitement is

So if all this is "old hat", why has interest been re-ignited? The answer is a combination of technology suppliers' interest in a set of application-level communication protocols (SOAP, WSDL, UDDI, WS-* and others) collectively known as Web Services technology; and industry interest as a whole in any cost-effective technology that might help ease enterprises' application development and integration burdens.

Web Services technology has added new colour to the SOA picture, because it has the potential to enable software integration and development to be carried out flexibly and rapidly, at multiple different levels of scope and abstraction – independently of the underlying implementation technology. Whereas most earlier middleware technologies provided development and integration flexibility benefits at the levels of individual compiled programs or possibly distributed systems, Web Services technology has the potential to glue software together across industry, organisation and geographical boundaries as well as across application, system and program boundaries.

SOA has business drivers, and it must be business-focused

Key industry developments point to a service-oriented IT approach

The challenges of operating in today's business environment

Globalisation of markets is driving customer choice to a point where competition for custom can be extreme; driving overall business success to depend not only on internal performance, but on performance of actions coordinated throughout complex value chains. Stiff competition also means that business success is often highly dependent on consistent customer service excellence – and therefore a customer-centric perspective which depends on the overall performance of the service delivered to the customer, which spans organisational functions.

The growing adoption of the outsourcing of IT, and increasingly of entire business processes, together with accelerating use of offshore service providers, demands the integration of these outsourced services with the rest of the organisation. At the same time, it must be possible to measure and enforce the quality-of-service and commercial aspects of third-party service provision.

The ongoing harmonisation of regulation, which is occurring against a backdrop of local regulatory demands, increases the burden of compliance. To alleviate the burden, it must be possible to assess the implications for business processes and map them to the IT services which underpin them. This demands flexibility to increase responsiveness and localise the impact of change, together with automation of the monitoring and reporting of the status of compliance.

Whilst the details vary, all these broad business challenges point to an increased focus on process improvement and specialisation, which in turn depends on improving IT-business alignment.

IT-business alignment points to service-oriented IT

The role of IT as a business support tool has changed significantly over the past decades but the thought processes of those involved in buying, selling and deploying IT have not kept pace, with the result that organisations' IT portfolios typically fall short. Information to support decision making is abundant but it is often difficult to access; new IT systems and applications are difficult to integrate; resource utilisation is sub-optimal with significant redundant capacity; and IT solutions are too inflexible to support business change. In short, it seems that very few enterprises' IT portfolios, strategies or delivery capabilities are well-aligned with the requirements of their business units.

Improving "IT-business alignment" requires an evolution both in technology and technology thinking, which makes it easier for organisations to more effectively exploit their existing IT assets; and re-prioritise IT investment, delivery and strategy so that IT assets can be more readily directed to deal with business change.

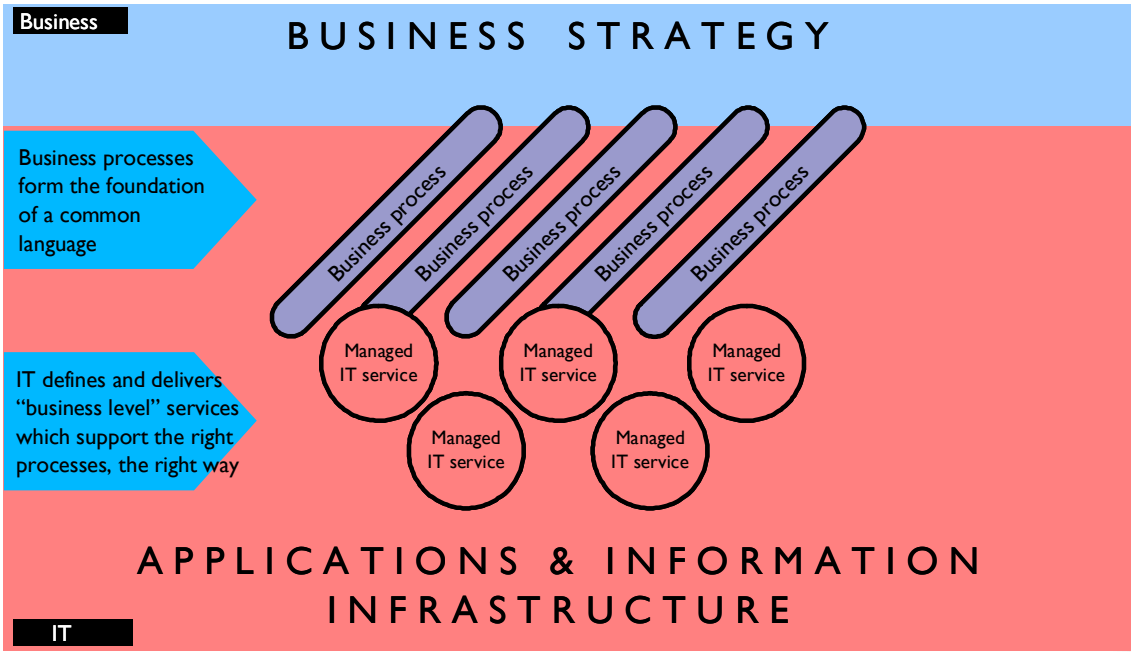
Figure 1 overleaf shows, at a very high level, an organising model for IT-business alignment.

In order to understand business change, there needs to be a common language to facilitate an effective, two-way dialogue between business and IT. Given organisations' need to focus on process improvement and specialisation, it is business processes which must provide this common language.

In support of business processes which are jointly understood by the business and the IT organisation, IT needs to adopt a service-based approach to the delivery of solutions, which is "business process aware".

This overall approach needs to be supported by a service-oriented model for enterprise software, which also enforces clear separation between underlying technology resources, the services through which they are exposed, the business processes which utilise those services and the means by which businesspeople interact with those processes. This is where SOA comes in – but SOA must be part of a wider service-oriented thought process.

Figure 1: Service-based delivery of IT



Service-based delivery of IT requires a common understanding of business processes and the definition and delivery of managed IT services to support these business processes.

SOA and the broader IT services spectrum

For any organisation serious about IT-business alignment, all IT, and all of the IT organisation's work, has to be concerned with service provision. As such, a service-oriented approach to IT must consider *all* aspects of that provision from a "service-oriented perspective" – not just those software-based services which are provided by business application functionality, but those which are provided by people, too. It depends on a way of thinking about IT services which makes no assumptions at all about:

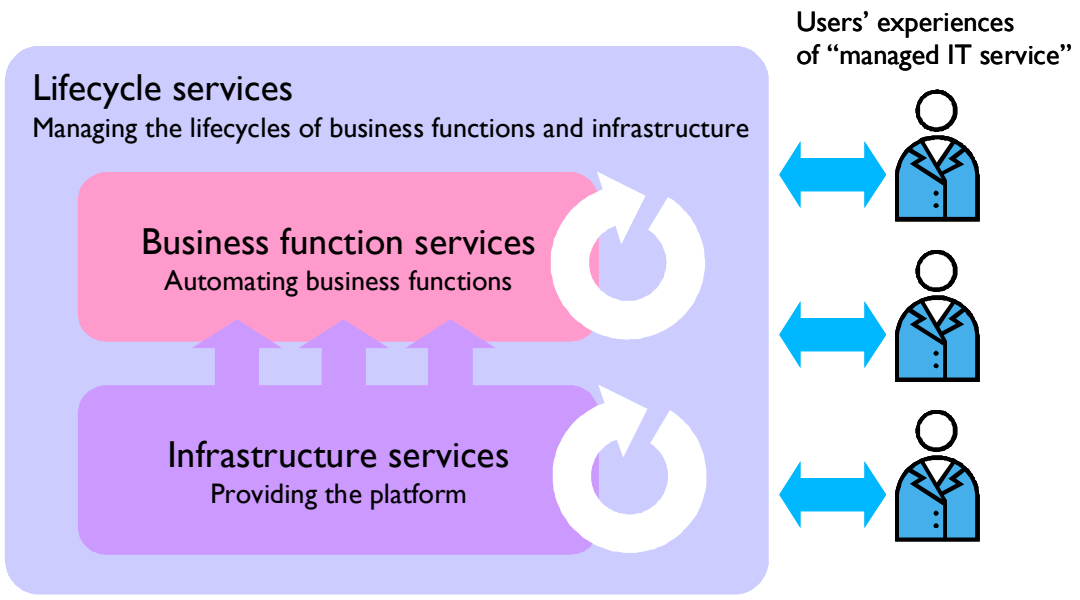
- Whether an IT service is provided directly by technology, or by people working with technology
- Whether an IT service is consumed directly by technology, or by people working with technology
- The "granularity" of an IT service (how "big" or "complicated" it might be)
- The different contexts within which an IT service might be consumed.

The spectrum: business function, infrastructure and lifecycle services

There are three types of IT service, as shown in figure 2, which between them cover all key aspects of the value that any IT service provider organisation offers:

- **Business function services**. It is these services which most people focus on, when talking about SOA. The "content" of these services is the core of what provides direct value to consumers, by automating aspects of particular business functions. Examples include customer history analysis, order entry, invoice processing, and product design automation. In the context of most enterprise IT scenarios, business function services will be provided to service users by business software applications or application components
- **Infrastructure services**. These services play a supporting role in delivering value to the ultimate service user, by providing the underlying platform over which business function services are delivered. Examples include data persistence management, authentication and authorisation, failover provision, communication, and auditing and logging. Typically infrastructure services are provided by computer and network hardware, and systems software of various types – databases, integration middleware, application servers, security servers, web servers, and so on
- **Lifecycle services**. These services are the "wrapper" which in the vast majority of situations provide the "real services" which business users of technology experience – they are responsible for managing the lifecycles of business function and infrastructure services. Primary types of lifecycle services are infrastructure support; software development; and application implementation. They may be provided internally, but often they are outsourced. In the vast majority of cases today these services are provided by IT professionals. Increasingly, though, aspects of how these services are delivered are being automated.

Figure 2: There are three IT service types, and they are inter-related



A service-oriented approach to IT should clearly define a set of managed IT services that encompass business function, infrastructure and lifecycle service elements.

SOA must be considered within the services spectrum if it is to add value

When business leaders set aside budgets for investing in IT, they are thinking primarily about the "lifecycle services wrapper" aspects of IT services. When software developers think about "services", they increasingly think in terms of exposing business software functions as "business function services". When IT operators think about "services", they tend to think about managing hardware, systems software and networks as "infrastructure services". Figure 2 shows that in reality, all these perspectives are inter-related – and only together do they enable the IT organisation to support business processes through a set of managed IT services, and therefore improve IT-business alignment.

The reason that this is important, is that it has clear implications for anyone looking at taking a service-oriented approach to application architecture. SOA initiatives will not deliver real business value unless they consider service-oriented application architecture in the context of the bigger picture of IT service provision: this means thinking in advance about the infrastructure implications of SOA, but – more importantly – the implications of SOA on current software lifecycle management practices.

The importance of contracts

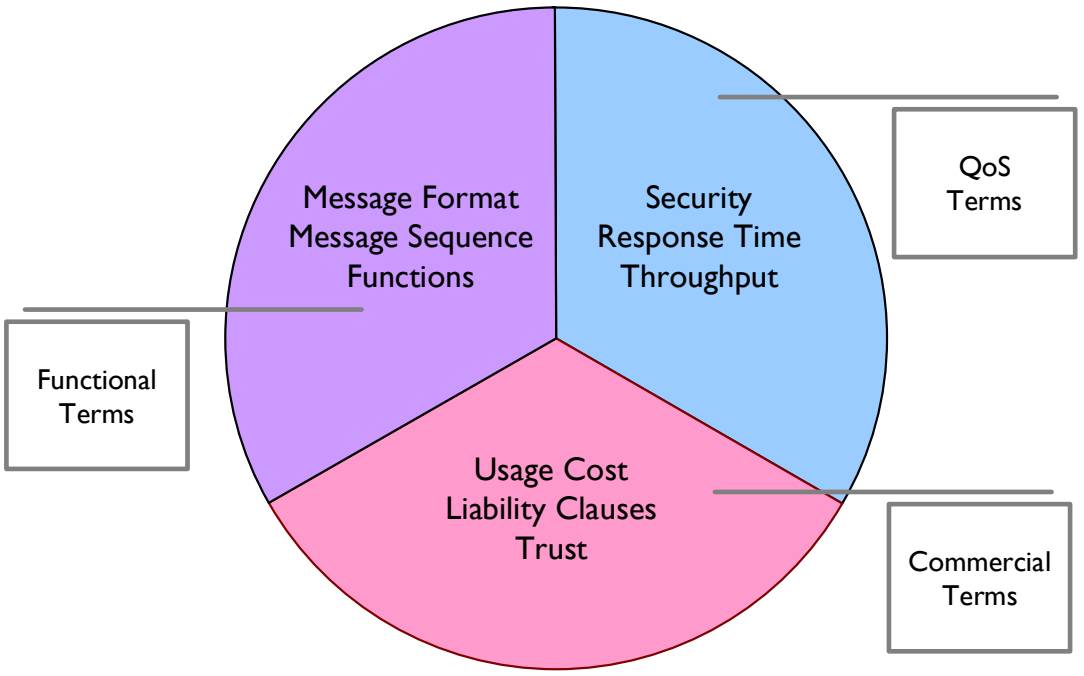
Irrespective of the (technology-based or human) nature of a given service provider and consumer, successful interaction between them depends on some kind of agreement on the parameters of the interaction. In a great many cases, agreements may not be formalised or even written down: but we can be sure that they exist – even only if in peoples' expectations. For a service-oriented IT approach to deliver real business value, it must get to the bottom of what these agreements and expectations are, and ensure that they are met.

Contract agreements for IT services define the obligations of the service provider (commonly including elements such as minimum response times, service availability, and data security guarantees); and of the service consumer (commonly including costs and payments, supported access device configurations, and guarantees about maximum numbers of concurrent users). An IT service contract is directly analogous to a contract in business terms, where it is used to define the mutual requirements and outcomes of service provision. In other words, a service contract should provide a precise and unambiguous definition of how the provider and consumer interact.

Just as we consider three types of service in the context of IT service provision, we also see three elements of contracts which govern interactions between service providers and consumers which are crucial to delivering business value – as shown in figure 3 overleaf:

- **Functional aspects**. These contract aspects specify "what" is provided: they dictate the functionality that the service provider must offer, and constrain how the consumer can invoke that functionality
- **Quality-of-service (QoS) aspects**. These contract aspects specify "how well" the "what" is provided: they dictate the operational quality that the service provider must offer (expressed in terms such as throughput, response time and so on), and constrain the terms under which that operational quality can be provided (for example, by relating response time to a maximum number of concurrent users)
- **Commercial aspects**. These contract aspects specify "how much" provision of the "what" and "how well" will cost: they dictate the price that the service consumer should pay, and how often; and specify the conditions that the service provider should meet in order to guarantee payment. Trust, privacy and security are important parts of these contract aspects: commercial services are invariably provided to particular, identified and "closed" customer groups which must not be compromised.

Figure 3: There are three aspects to IT service contracts



Functional aspects of IT service contracts may be supplemented with aspects which govern qualitative and commercial characteristics of the interaction between provider and consumer.

Critical success factors for SOA initiatives

The preceding discussion defines a way of thinking about SOA which is business-focussed. Here we describe the critical success factors that, if addressed, will ensure that SOA initiatives are much more likely to deliver tangible business value and provide a demonstrable contribution to IT-business alignment.

Implementation factors

Application architecture is at the core of SOA, and initiatives should therefore start from the perspective of business function services. However, as we have already demonstrated, this perspective needs to be adopted in a broader context. This context must encompass more than functional contracts; consider the relationships between business function services and other types of IT services; and consider the business processes that business function services must support.

An incremental approach which starts with business processes …

The starting point for business function service design should be an understanding of the nature of business processes and how business function services have to support them. It is important to understand that business processes are not all the same: processes can be fundamentally different from each other in terms of the role they play in the organisation – and this affects how you must consider the definition of business function services. In particular:

- Some business processes serve to differentiate the business from its competitors, and "make the business what it is"; whereas others are merely "the cost of doing business" – they are necessary, but they play no role in helping to really define or differentiate the business
- Business processes act at different levels: many are concerned with day-to-day business execution; but there are other processes, not so often analysed or talked about, that concern business management and strategy. "Higher level" business processes tend to be less structured and predictable than processes at lower levels
- Business processes are automated (and automatable) to different degrees. Processes concerned with business management and strategy are unlikely to be supported by IT in a structured way, let alone automated; whereas processes more concerned with day-to-day business execution, and processes which are more stable, structured and mature, are more amenable to automation.

The design of a structured portfolio of business function services should draw on top-down analysis of the services required to support the different types of business processes, together with a bottom-up analysis of existing systems.

The outcome should be a hierarchy of business function services, which maps to the hierarchy of business processes that exist in the organisation where SOA is to be applied as a supporting architecture.

Business function service design must also resolve the well-recognised trade-off between software flexibility, openness, reusability and efficiency. This trade-off likely to be felt particularly keenly in the context of SOA, because openness and flexibility of access to software services are so often considered to be "done deals", because SOA is so often considered in the context of Web Services. In reality, though, it often makes sense to place limits on the openness and flexibility of software in order to make things as efficient as possible. Similarly, pursuing a design strategy which blindly maximises the potential reusability of resources, would yield a multitude of fine-grained services which can not be reused in practice.

By starting the design process with a thorough understanding of business processes and how they need to be supported by software, addressing this trade-off becomes less of an art and more of a science. For example, the demand for flexibility and openness is greater in support of business processes which are differentiating; or which act at strategy and management levels. In contrast, efficiency is often a more important consideration in support of business processes which are concerned with day-to-day business execution, and which are "non-differentiating". Similarly, reuse of business function services becomes more important at the lower levels of the business process hierarchy.

The challenges faced by the enterprise IT organisation in defining their business function service hierarchy mirror many of those faced by commercial software development organisations. In the face of this daunting prospect, organisations should use their understanding of business processes to identify "service domains" that map to high-level business activities (for example, product and service creation; sales and marketing), which share common semantics and policies. Effort can then be focussed on particular domains based on the relative priorities associated with them, enabling an incremental approach to SOA.

… considered in a broader context

It is easy to fixate solely on functional contract aspects – after all it is these which are easiest to consider, given the application development centred focus of technology vendors currently offering "SOA solutions". Nevertheless it is crucial to move beyond functional issues, and consider the other contract aspects which contribute to the transformation of business software into business function services:

- **Quality of service aspects** – how can consumers ensure that business function services exhibit the QoS levels that they expect; and how can providers ensure that consumers don't make demands that are outside the boundaries of the service parameters that have been agreed?
- **Commercial aspects** – how can consumers ensure that providers meet their overall service obligations for all key business function services; and how can providers and consumers mutually ensure that the environment in which they are operating is trusted?

In addition, it is crucial to consider not just what it means to transform a piece of business software into business function services: but also the implications that moving business function provision towards a more modular, distributed, open runtime environment has on infrastructure services and lifecycle services:

- **Infrastructure services** – given the runtime accessibility and reusability of business function services, how can providers ensure that infrastructure services can support the quality-of-service needs of individual business function consumers, each of which might have a different requirement? How can the operational risks of a much less predictable software usage environment be minimised? How can access to business function services be secured in an environment that spans traditional domains of control? How can providers ensure that consumers gain meaningful insight into the operation of business function services?
- **Lifecycle services** – how can changes to business function services be best prioritised and managed? How can the need for rapid change in some cases be reconciled with the need for high-quality reusability in others? How can dependencies between business function services be understood? How can the potentially conflicting needs of different consumers be reconciled?

Anticipating complexity

Our categorisations of IT services (business function, infrastructure, and lifecycle services) and contract aspects (functional, quality-of-service, and commercial aspects) should be used as an organising model for thinking about architecture, as it yields valuable insights into the issues that SOA should be addressing. Without such a model to help locate SOA in the larger landscape of IT-business alignment, SOA is in danger of becoming another footnote in the history of failed distributed computing phenomena, stifled by complexity.

Figure 4 shows how thinking about the interactions between IT service types and contract aspects helps to anticipate and manage that complexity. Each cell of the table is used to illustrate the implications of a particular contract aspect, when applied to a particular type of IT service – for example, the centre cell in the table illustrates the concerns which arise when quality-of-service contract aspects (response time; availability; reliability; and so on) are applied to infrastructure services (database management; transaction processing; user authentication; and so on). Figure 4 only provides examples of the kinds of issues that arise: a detailed discussion is beyond the scope of this particular report.

Figure 4: Example architectural considerations in the context of particular IT services and contract perspectives

	Functional contract aspects	**Quality-of-Service contract aspects**	**Commercial contract aspects**
Business function services	How can we align flexibility / reusability requirements for business software functions, to business needs?	How can we trade-off flexibility / reusability of business function services with efficiency / openness requirements?	How can we ensure that the right consumers get the right kind of experience from these services, and do so cost-effectively?
Infrastructure services	How should infrastructure elements provide their services to different business functions? How should infrastructure be optimally managed?	How can QoS responsibility be delegated to infrastructure in a way that is easily flexed in response to changing requirements?	How can we minimise the cost and risk of overall process support while creating an business support environment with more moving parts?
Lifecycle services	How should we differentiate lifecycle service levels for different kinds of business function, infrastructure?	How should we define and enforce development, fault-fix and change-request priorities?	How can we demonstrate the overall value of the services that the IT organisation and its resources provide to the business?

Particular kinds of IT services, and particular contract aspects related to those, drive different architectural considerations. Each perspective gives rise to questions which have to be considered in the context of related perspectives: this is a world of tradeoffs.

Technology factors

Having discussed a number of the implementation factors that are critical for the success of SOA in the context of service-oriented IT initiatives, this section details a number of the things you should look to technology providers to support you with.

An integrated approach to lifecycle management

The hierarchical nature of services, coupled with the potential for service reuse increases both the likelihood of change and, as a result of the greater inter-dependency of services, the potential impact of change and the complexity of assessing its impact. These factors also demand an increased focus on service quality, particularly at lower levels of the service hierarchy. Mastery of service version and change management, which can (for example) enable multiple versions of services to be deployed and operated in parallel to support the needs of particular consumers, is therefore critical in any non-trivial deployment of software services.

An integrated approach to the management of the service lifecycle, which focuses on ensuring appropriate levels of service quality, and enforces consistency across the service portfolio is key to mitigating these risks. At the same time, it provides a means for effective control of the service portfolio.

Policy-driven quality-of-service definition and assurance

As we explained in our introduction to contracts above, successful interaction between a service provider and consumer depends on satisfaction of consumer and provider obligations as defined in service contracts. Openness, flexibility and reusability are the key goals of a service-oriented approach to IT, which raises the question of how these contracts can be enforced in operational systems in a manageable way? By encouraging openness, flexibility and reuse, a service-oriented approach guarantees that we cannot know in advance which consumers might request which services. We cannot know what kinds of obligations might need to be fulfilled, until a request is made.

The way to handle this uncertainty, which is sure to arise as service portfolios expand and become more complex, is to use the design concept of "policy" to dictate the conditions which must exist for the contract between a consumer and a provider to be fulfilled.

It is important to realise that the policy enforced for a particular interaction (dictating, for example, the security credentials which are required for consumer authorisation) might depend on:

- The identity of the consumer
- The identity of the provider
- A combination of consumer and provider
- Some other information about the context of the interaction (for example, the value of a requested transfer from one bank account to another).

These "service instance policies" are the mechanism through which responsibility for the quality-of-service aspects of contracts can be delegated to infrastructure. A service-oriented approach to the delivery of IT demands intelligent infrastructure, capable of assuring quality-of-service based on declarative policies. A number of technology vendors have ambitious visions to address this need.

These visions, of course, remain just that and are some way from being realised. In the meantime, quality-of-service contracts are still an important tool in the architecture of service portfolios. They provide a means for manually defining policies which can be enforced by a range of technologies, including systems management tools, message oriented middleware and specialist web service routing brokers, and federated identity management solutions.

However, the increasing maturity of service portfolios which will come as service-oriented approaches become more common within organisations, will stretch these manual approaches. With greater proliferation, it will become increasingly costly and time-consuming to manually define policies for multiple services in multiple contexts.

A metadata management core

It is clear from the preceding discussion that the success of an SOA initiative depends on the effective management of data about IT services. Service metadata management facilities must extend beyond a simple service catalogue providing access to business function services and associated functional contract aspects. They must also encompass the quality-of-service and commercial aspects of contracts to facilitate policy-driven service assurance and to underpin business-relevant IT governance models.

The complexities of real-world interactions between deployed software services, particularly given the potential for services to interact across organisational boundaries, mean that hard-wiring authentication, authorisation and usage tracking characteristics to particular services, is going to be too inflexible and costly. As a result, there is a growing realisation of the important role that identity management has to play in SOA initiatives. It is therefore important that service metadata management facilities are integrated with identity management solutions.

A call to action

The real opportunity – and challenge – of SOA, comes from considering a simple question: "what is an IT service, exactly?" The answer is "it depends" – and it is the ability of Web Services technology to unite the multiple different perspectives which arise from this consideration, that truly gives SOA the potential to transform the ability of IT organisations to support business needs.

However, the vast majority of current "service oriented thinking" concerns the evolution of application architecture. But to add real business value, SOA cannot be just about application architecture: it must also enable a planned and systematic approach to delivering service-oriented IT.

You must move beyond the common understanding of SOA that permeates much of today's thinking (a particular arrangement of middleware technology which provides infrastructure for integrating and composing web services). Whilst pursing an IT initiative based on this understanding can bring some benefits, it will not by itself deliver significant improvements in the alignment of IT with business needs.

In order for an SOA initiative to really improve IT-business alignment, you have to consider it as part of a larger picture, which encompasses and links different perspectives of IT services – a true picture of "service-oriented IT", which puts managed IT services at the heart of the relationship between business and IT. The vast majority of today's SOA initiatives focus firmly on one type of IT service: *business function* services (software services which automate aspects of particular business functions). But there are two other types of IT service which need to be considered: *infrastructure* services (which provide the underlying platform over which business function services are delivered, and which tend to be the focus of IT operations professionals) and *lifecycle* services (which are responsible for the design, implementation, operation and alteration of infrastructure and business function services, and which tend to be the focus of business investment in IT).

This big picture view of "service-oriented IT" should not, however, be taken to imply a big bang approach. Instead, you should use your understanding of the different types of business processes, their relative priorities and domains of business activities to drive an incremental approach, supplemented by a bottom-up analysis of your existing systems. This approach must take account of not only the functional aspects of the contracts between service providers and consumers, but also the quality-of-service and commercial aspects which contribute to the transformation of business software into business function services. This understanding also provides a solid foundation for business-meaningful IT governance.

In addition, you need to consider not just what it means to transform a piece of business software into business function services: but also the implications that moving business function provision towards a more modular, distributed, open runtime environment has on infrastructure services and lifecycle services.

From a technology perspective, you need to recognise the importance of integrated lifecycle management to ensure quality, consistency and control. You must also move to a policy-based approach to quality-of-service definition and assurance to dictate the conditions which must exist for the contract between a consumer and a provider to be fulfilled. These must be underpinned by a very solid metadata management foundation, which extends beyond a simple business function service catalogue to encompass the other aspects of service contracts and integrates with other enabling infrastructure services, particularly identity management.

About Systinet Corporation

Systinet provides the foundation for SOA governance and lifecycle management, making IT simpler, faster and standards-based.

With its suite of award-winning and proven products, Systinet enables organizations to rapidly leverage and reuse their existing applications and data assets, provide interoperability among heterogeneous systems, and better align business processes with IT. Systinet's products enable, publish, discover and manage SOA business services, provide robust policy management, and make it easy to build secure and reliable Web services with Java and C++ applications.

Systinet products are based on industry standards such as XML, SOAP, WSDL and UDDI. A pioneer in SOA technology, Systinet led the development of important standards at the World Wide Web Consortium (W3C), OASIS and elsewhere, while remaining consistently first-to-market with advanced and innovative products based on these standards.

More than 160 Global 2000 companies rely on Systinet technology, including Amazon.com, BT, The Hartford, Interwoven, JPMorgan, Motorola, Defense Information Systems Agency, and Société Générale. Founded in 2000, Systinet is a privately held company with headquarters in Burlington, Massachusetts.