



# White Paper

## Service-Oriented Application Management

*Application management was initially pretty operational and reactive. About fifteen years ago, in response to the higher service levels that the user organisations required, we started investing in improvement of the maturity of application management processes. We are currently slowly moving towards a new phase in which we're looking for an adequate response to the increasing complexity of a new generation of 'virtual' applications based Service-Oriented Architecture (SOA). The areas that require extra attention are architecture, ownership (governance) and an 'adhocratic' kind of organization.*

Mark Smalley, 10 April 2008  
mark.smalley@aslbisfoundation.org  
www.aslbisfoundation.org

# SOA and IT Management

## SOA is sexy

Applications used to consist of several large programs that were difficult to modify and that were interconnected with batch-interfaces or by sharing data in common databases. These monolithic applications were custom made and encompassed the various architectural layers: presentation, business logic and storage. The two-tier and multi-tier client/server application that appeared in the eighties and nineties were generally more modular and distributed and made use of graphical user interfaces and back-end database and application servers. SOA evolves the client/server concept a step further but retains the proven design principles of *loose-coupling* and *tight-binding*.

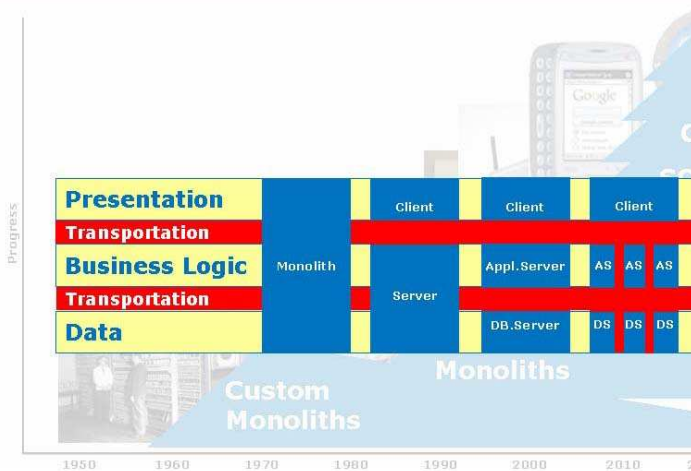


Figure 1: Generations of application architectures

The term 'SOA' was coined by Gartner in 1996 but it took ten years before serious (but small-scale) adoption occurred. The development of standards such as HTTP, URI, XML, SOAP and WSDL played an important role in lowering the barrier. Adoption of SOA - if done well - results in the speeding up the realization and implementation of changes in the functionality. The development of the first SOA application for an organization will take as long - if not longer - than a non-SOA application. This is because of the investment in specific architectural and infrastructural facilities such as an Enterprise Service Bus and Service Registries/Repositories. The improvement will be noticed when developing other applications and when implementing changes. SOA can be used in order to connect applications with differing technologies. Unlocking the value of legacy applications is a very attractive prospect for many organizations.

This technology push is supplemented by a business pull from the user organization. Organizations experience the pressure of having to adapt rapidly to changing circumstances. This occurs for example in acquisitions and when their position in the supply chain occurs, especially when the supply chain partners have to exchange more information. SOA is promising and is therefore a 'hot topic'.



## IT Management is boring

In order to take a good look how service-oriented applications can be managed, it's useful to realize how IT management has evolved. Fifty years ago we only talked about 'computing' or 'EDP', without a further breakdown in discipline X or Y. As Professor Maarten Looijen said, "everybody did everything". Then our attention focused on systems development: since there were few systems, they had to be developed. Anything other than 'development' had no specific name, you just did it. Only later, in the eighties / nineties, we made a distinction between 'application management' and 'infrastructure management'. Until this point the user organization didn't have any role of importance but a few years later this changed and we started talking about 'business information management'. This, in a nutshell, is the succession of paradigms which helped us understand IT management. Currently we use the subdivision in business information management, application management and infrastructure management, supported by process models such as BiSL, ASL and ITIL.



Figure 2: IT management paradigms

The image of IT management has undergone a true metamorphosis in the last decade, from reactive and operational towards tactical and here and there even strategic and proactive. But this improved image pales into insignificance when compared to a bling-bling technology such as SOA.

## Management of SOA applications is risky

One of the foundations of IT management is having an overview of the components of a system that need to be monitored, changed etc. The functional design of a program is modified. Software sources are coded. Object code is installed in the production environment and its performance is monitored. Ditto for servers, operating systems, networks and so on. These are examples of type of artifacts. For management of traditional applications we could suffice with a list of the programs and data structures that comprised an application. Usually the application management department acted as 'owner' of these artefacts.

SOA changed lots of this. The application is now composed of components (services) that may be owned by the application management department and/or - and increasingly - by other departments or third parties. This alone is a new phenomenon for the application management department that is accustomed to have everything under its own control. Now there is a



dependency on others, with which operational level agreements have to be made. And there are new types of artefacts to deal with, including:

- Services that are invoked (and that can also invoke other services)
- Applications that invoke other services, providing users with the information they need
- Directories with a list of the available services
- Brokers that execute the actual invocation

In addition to these new artefacts, there are also new activities, such as:

- Service Discovery, the process of finding services with the desired properties, with the challenge of maintaining the list of services in such a way that optimal use is made of existing services rather than accidentally creating duplicate services
- Service Composition, with special attention for the integration of the various services into an application; input/output conversion, and coordinating transactions
- Service Invocation, with special attention for the actual running of applications and invocation of associated services; ensuring adequate availability and performance

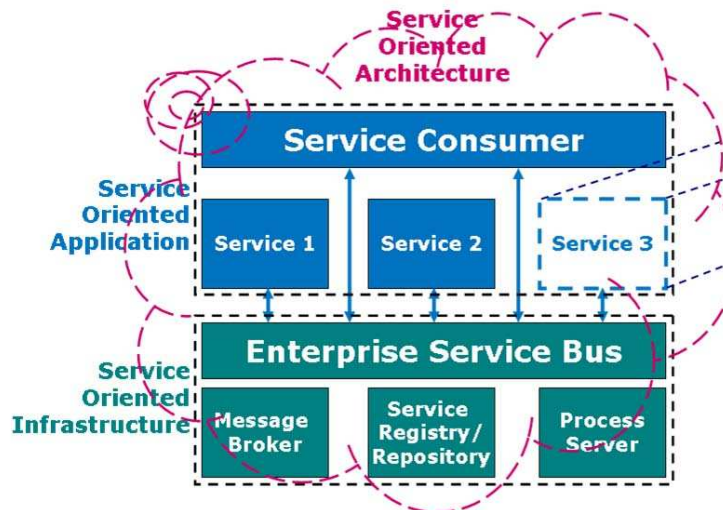


Figure 3: Types of artefacts

In addition to the application artefacts mentioned above, SOA introduces new artefacts in the infrastructure domain: Enterprise Service Bus, Message Brokers, Service Registries / Repositories, Process Servers. As these are new components for the system administrators, some significant training is required if risks are to be contained.

The sketch of these changes is not complete but illustrates several differences with traditional applications. Management of SOA applications is a new area with new types of artefacts that must be managed, with new activities to be carried out and new concepts that have to be learned. So there's plenty to do, but how to approach it?

# Recommendations

## Adhocratic approach

As mentioned before, it's wise to look back at where we've come from before taking a step ahead. We've often experienced the passage for one technology to another. What can we learn from this? The development of each new technology follows an S-curve. At the beginning of the S-curve it is unclear exactly how to tackle the new technology but in the course of time the specialists get a grip on the matter and are able to formalise the processes, giving a feeling of 'under control'. People who are competent in an established technology - for example in the mainframe domain - and are therefore at the end of their S-curve, look with stupefaction at how their (frequently younger, that's the way it goes...) colleagues wrestle with the new technology. The proven tooling, procedures, training etc that the old technologists use, has not yet been created for the new technology, making it less credible from their perspective.

New technology S-curve:  
one step forward,  
two steps back

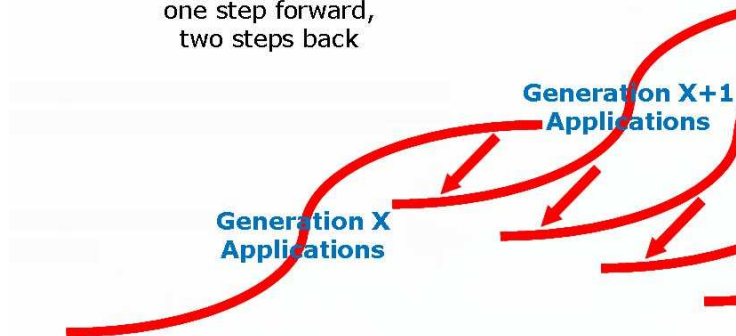


Figure 4: S-curves

We have encountered this phenomenon around the turn of the century, with the advent of internet applications. The regular IT departments were uninterested in - or scared by - a new technology that they regarded as a plaything so the business took the initiative and created special Internet teams outside the IT department. These teams were staffed by business-representatives, software specialists, network specialists etc, closely cooperating with each other in order to achieve a workable outcome. Bit by bit the process was formalised and in the course of time it was 'promoted' to normal IT, executed by the IT department. This multidisciplinary approach with highly knowledgeable specialists, unclear roles, intensive communication and informal coordination is known by Bennis, Tolfer, Mintzberg and Waterman as 'adhocracy'.

When managing SOA-based applications, it is important to realize that we starting on a new S-curve, and that we should realise that we lack knowledge and that we therefore have to go through a learning curve before it the process can be formalised . An approach with a multidisciplinary team (centre of excellence, competence centre) currently seems the best approach to take.

In addition to this recommendation to adopt an adhocratic approach, investment in two other areas is strongly suggested: architecture and ownership.



## Architecture

It may seem paradoxical, but a system that is based on extensive loose-coupling requires some form of central coordination. If not, you run the risk that the intended benefits of SOA will not be realized. Also, the failure of a SOA initiative will lead to a large divestment, not only financially but also in the reputation of the IT department.

A quote from some research performed by the University of Amsterdam, supporting this assertion: "Without professionals who work according to architecture principles and without management of the architecture processes, the probability of success for SOA is very small."

## Ownership

Another quote from the University of Amsterdam: "Regarding larger applications of SOA, there is some resistance and reluctance surrounding ownership of services."

A SOA-based system consists of various components, many of which are used by other systems and departments. This complicates governance and financing issues. The question of who decides about what should be settled in advance. Decision-making about and use of common components should also be addressed. Suppose a division makes a component available for use by other departments and incurs extra costs. How will this be funded? If this is not sorted out, there is a significant risk that the intended benefit from reuse will not be achieved and that suboptimalisation will occur, with different versions of similar services being created.



## Conclusion

Application of SOA principles promises more flexibility, better integration and lower costs. However, the advent of such applications has implications for how application management. If SOA-based systems are not effectively managed, there is a considerable risk that the intended benefits - flexibility and integration - will not be achieved. The knock-on of this is that organisations will not be prepared to continue to invest in this technology, or that the investment will be spread out over a longer period, implying that the benefits will not be reaped as soon as they could be. Paying attention to the following areas - architecture, ownership and the manner of organization - can reduce these risks.

## Architecture

More than with traditional applications, attention should be paid to the use of overarching architecture principles, otherwise there is a risk that the intended benefits of SOA will not be reaped. Also, the failure of a SOA initiative will lead to a large divestment, not only financially but also in the reputation of the IT department.

## Ownership

A SOA-based system consists of various components, many of which are used by other systems and departments. This complicates governance and financing issues. The question of who decides about what should be settled in advance. Decision-making about and funding of common components should also be addressed.

## Organisation

Finally - because management of SOA-based applications is currently immature and therefore still difficult to formalise - it is prudent to manage such systems in an 'ad hoc' way (multidisciplinary team with high skilled specialists, intensive communication and informal procedures). Once best practices have been created, the processes can be formalised and it can be 'promoted' to normal IT in the regular IT department.

## Author

Mark Smalley is specialised in Application Management and works as Principal Consultant for Getronics PinkRocade, also representing the ASL BiSL Foundation in international affairs.

## Reading list

M. Looijen, in ICT Pocketbook, Reed Business Information, 2007, pages 383-389.  
M. Smalley, Integrating in the ICET phase! in IT Service Magazine, 12-2007, pages 18-21  
Key Issues for SOA, EDA and WOA, Gartner, 2008  
Introduction to Service-Oriented Architecture, Gartner, 2003  
Migration of Legacy Assets to SOA Environments, Carnegie Mellon Software Engineering Institute, 2006  
SOA Trend Survey, InfoWorld, 2006  
SOA in Nederland – a digital disease?, University of Amsterdam, 2007