

2 section two

An Incremental Strategic Approach to Service-Oriented Architecture



Today, more and more organisations are turning to a Service-Oriented Architecture (SOA) to increase productivity, enhance operational efficiency and agility, and align their IT infrastructures with business strategies. So what is a Service-Oriented Architecture?

SOA is an application architecture, in which application components or "services" are well defined using common interfaces, utilise a contract to define how services will be invoked, and interact in a loosely coupled manner. In this paradigm, the terms "client" and "server" are purely situational. At one moment, an application could act as a client by calling an external service, while moments later, it may act as a service-provider when called by another application to perform a task. When properly implemented, SOA promises to end the building and maintenance of point-to-point integrations. Using SOA, businesses will be able to generate new services in a flexible and agile way by combining existing logic and exposing it via reusable services.

Ivo Totev
Chief Marketing Officer
Software AG HQ

Service-Oriented Architectures should be Introduced Gradually and Systematically

1 Introduction

Today, more and more organisations are turning to a Service-Oriented Architecture (SOA) to increase productivity, enhance operational efficiency and agility, and align their IT infrastructures with business strategies. So what is a Service-Oriented Architecture? The SOA landscape is strewn with buzzwords, conflicting technologies, architectures and pitfalls. There's no single approach or "solution". So exactly what is an SOA - and what does it provide?

SOA is an application architecture, in which application components or "services" are well defined using common interfaces, utilise a contract to define how services will be invoked, and interact in a loosely coupled manner.

In this paradigm, the terms "client" and "server" are purely situational. At one moment, an application could act as a client by calling an external service, while moments later, it may act as a service-provider when called by another application to perform a task. When properly implemented, SOA promises to end the building and maintenance of point-to-point integrations. Using SOA, businesses will be able to generate new services in a flexible and agile way by combining existing logic and exposing it via reusable services.

2 SOA and Web Services

SOA is not new. In fact, technologies like CORBA and DCOM promised SOA in the 1990s. Both CORBA and DCOM introduced a high degree of complexity, and services based on these protocols were not truly interoperable by being vendor-dependent. Thankfully, today XML and Web services offer a truly standards-based approach to developing services that can be consumed by many different types of applications—regardless of the application development technologies at work.

It is important to note that not every SOA is based on Web services. Rather, many different technologies, protocols and standards can be at play within an SOA. Web services are but one example of a standards-based implementation.

3 Making SOA a Success

Reuse is one of the motivating factors of SOA. But what does reuse mean in the context of SOA? Does it imply the reuse of only newly created Web services? Does it also include the reuse of existing applications? Does it include the reuse of existing programming and technology best practices, and/or software development guidelines? Also, how do we achieve reuse?

Many organisations that have embarked on the SOA path have come to realise that achieving true reuse is not as simple as creating Web services and making them available for consumption. Many different factors complicate the simple act of using services that have been made available within an enterprise.

Understanding the nature of reuse within the context of SOA and how to generate the adoption of services are key steps toward realising the promise of SOA.

4 What is Reuse?

Let us take a moment to explore what reuse means in the context of SOA. Reuse can take many different forms. A simple definition of the term is the ability to repeatedly use the exact same service, component, procedure, guideline or process by various people or service consumers in order to fulfill a given task. From this definition, it is clear that reusability extends beyond the mere reuse of Web services or existing applications exposed as Web services. This definition also implies other focus areas and disciplines that build a culture of collaboration—which provides the means to create true services reusability.

5 Reuse Hurdles

Reuse of services is often more complicated in practice than in theory. There are many reasons for this. The most frequent problems include:

Lack of organisational support: You need the appropriate support from both architects and management when implementing an SOA. The biggest challenge is not always related to technological issues. The biggest challenge is usually the business culture.

Implementing an SOA implies change, sometimes lots of change, and people deal with change in different ways. To some extent all of us are resistant to change - especially when we have not been included in a decision that impacts us or when we might not agree with a given approach.

Many organisations find that to successfully implement an SOA and get the appropriate levels of reuse, they need firm support from all levels within the IT organisation. Management support is most crucial since management drives the IT strategy, provides funding for these kinds of initiatives and provides backing when important decisions have to be made.

Likewise, without the support and participation of IT architects, managers and development teams, an SOA initiative is likely to fail. It is much easier to achieve SOA reuse goals when development teams are committed to sound strategy and follow established architectural guidelines and best practices.

Lack of guidelines and best practices: Many SOA implementations struggle with the lack of focus and conflicting agendas. That's why you need to establish a clear SOA discipline with pragmatic best practices and programming guidelines. Most programmers learn to create services by example. The most effective way of helping programmers adopt Web services and SOA is by providing them with practical examples and guidance on how to create and consume Web services. At the same time, you have to help programmers avoid common mistakes and bad programming techniques that cause interdependencies and interoperability issues.

Interoperability: Interoperability issues make it difficult for consumers to interact with service producers. Often a consumer can interact with a Web service that uses the RPCbinding style. Yet the consumer cannot interact with a similar Web service that exposes itself using a Document/Literal binding style. Sometimes consumers are unable to deal with complex data types created by exposing a legacy application. This is an area where standards such as WS-I help a lot.

Developers can prevent such issues by establishing programming guidelines and best practices and by following established architectural blueprints. By anticipating interoperability issues, the architectural team can put a framework in place to address these incompatibilities if they arise.

6 Getting Started with SOA

Converting to a service-oriented architecture (SOA) requires close collaboration between different company departments and its IT group. Prior to the project start, it is essential to find employees who, in the role of business analysts, can grasp technological requirements and translate them into technical specifications for the IT group. Likewise, architects must ensure that components are reusable and a uniform technological basis is developed over the course of the project.

Here you will read about..

- How IT decision-makers can meet the challenges of introducing and operating an SOA;
- Which phases are included in the SOA life cycle; and
- How to master the increasing complexity of an SOA.

The SOA life cycle begins with the discovery phase. During this phase, the project team first identifies the most important core usable business functions within the company and sets the project's goals in conjunction with management. The managers should first indicate how selected business processes could run more efficiently through the inclusion of central IT services. Then, in the assessment phase, the concrete business processes are identified. In practice, this often leads to a business reengineering project supported by external consultants. The process can be accelerated if available process documentation can be used.

7 The Search for Business Services

Defining the SOA's business services is crucial to the success of an SOA implementation. Implementation is only successful when the SOA team does not deal with the architecture and its underlying technology, but primarily ascertains which services a company really needs. It is easier to calculate the return on investment if the need is known and concrete benefits can be shown.

8 The Bottom-Up Approach

One example of a business service is company-wide customer data management. Here, the service provides a central interface that changes, deletes, or queries customer information in accordance with the content of the request. This bottom-up approach leads quickly to initial successes, because a concrete business problem is solved with available IT systems. The automotive industry serves as a model - the platform concept is long-established here. Manufacturers have identified central components that can be used - slightly modified if necessary - over and over again in new vehicle models. Organisations that would like to implement an SOA are pursuing a similar goal.

The time needed for the first two phases is clear. Two to four days are needed for the kick-off meeting with the project team, and three to four weeks for the subsequent assessment phase. The result is a clear description of the services and processes that are most commonly used within the organisation. For the implementation it is then necessary that the company introduce uniform interface technology - for example, Web services. Only in this way can services be reused in different scenarios in accordance with an SOA concept. The legacy integration and modernization tools used for this purpose must be able to cover all systems available within the company. Otherwise, a two-class society with both linked and isolated systems will exist within the SOA world.

9 Ten Rules for an SOA Project

- 1) Note that: "Organisation is just as important as technology."
- 2) Resolve communication issues between IT and departmental managers, with external mediators for example.

- 3) Do not implement SOA management and governance as an afterthought; ensure support for the complete SOA life cycle in a timely manner.
- 4) Start with small projects as a way to make iterative progress.
- 5) Treat all IT systems and technologies to be integrated into the SOA the same.
- 6) Reduce dependencies between individual services; they have an impact on reusability.
- 7) Process optimisation is a key theme for SOA, and business-process management is a key component.
- 8) Ultimately users see only an SOA's user interface. Therefore, base your development on such user friendly technologies as Ajax.
- 9) Use technologies for legacy integration that allow bidirectional data exchange with other services.
- 10) Create the position of SOA librarian, serving as an information centre.

There are several approaches to integrating existing systems on mainframes: direct access to the data or indirect access via the relevant application. In the context of SOA, the second approach is often preferred because existing business logic and integrity rules can be used.

10 Simulate User Dialogs

Ideally an existing system is already structured with individually callable functions, which are suited for use as services in an SOA platform and need only be "wrapped" in the respective standard interfaces. In practice, however, many systems are monolithic and have developed over many years. Reengineering these systems into modular services is costly and not without risk. Simulating user dialogs presents itself as an option for the transition period and for less strategic applications. When the source code or the programmers who are familiar with it are no longer available, integration via user interface becomes the only way to integrate legacy systems into an SOA world. The result of this phase of SOA enablement is a number of fine-grained components.

11 Composition, Orchestration

The leverage phase that follows in the life cycle consists of packetising the truly necessary services with the help of a composition and orchestration layer, so that all IT functions that employees use for technical processes are provided in the SOA platform. Functions can be composed into a new business service via an enterprise service bus for instance, so that users must no longer access different existing systems or screens. Instead, they are presented only with the result of their inquiry. In addition, it is necessary to implement business services so intelligently that they decide dynamically, depending on the content of a query, how to serve each kind of back-office system.

12 Ajax Helps at the Front End

Model-based tools generate orchestrated composite applications from these services. Composite applications are a new kind of application, characterised, for example, by a high degree of flexibility and having interactive user interfaces with modern Ajax (Asynchronous Java Script and XML) technology. If services that can be accessed by the public, such as Google Maps or Amazon.com, are integrated into an SOA system, a new generation of open business applications is created: mashups. These applications run without being installed in the browser and behave interactively like a desktop application. Ajax technology is well suited for implementing such applications. Developers should refrain from manual Ajax programming, however, because of its high degree of complexity, and instead, use frameworks that generate and administer Ajax on the basis of visual models.

If composite applications must support several roles and process steps, companies should implement business process management (BPM) systems. Using them, they can obtain a business view of the processes. Thus, for instance, all activities needed to answer a customer inquiry can be modelled and automated. The necessary connection to the IT systems is created by linking the business services with the BPM system.

13 Granularity Options

The number of components within a business processes that can be merged into a business service via an enterprise service bus depends on their granularity. Determining granularity is a matter of experience. Nevertheless, there are two boundary conditions that can be helpful. If the response times for an application based on SOA components are good, but the reusability of the components is severely limited, then the services are too coarse-grained. On the other hand, if the services' reusability is good, but the performance not, then the system is too fine-grained. Companies can find support for fine tuning through consultants or manufacturers who have experience with the implementation of SOA projects.

14 The Five Biggest Pitfalls

- 1) SOAP and Web services are not a panacea for basic architecture problems.
- 2) SOA consists of distributed systems and cannot be run according to the same rules as a monolithic system.
- 3) Avoid big "waterfall model" project introduction. Start small and work iteratively.
- 4) Rip and replace does not fit the SOA model. All existing relevant systems should be integrated into the new architecture.
- 5) SOA encroaches upon workflows. Thus, managing business processes is an essential task within an SOA project.

It is important in this context that project managers look at the bigger picture of IT implementation and make sure that success at the business level is measurable. This can take place on the basis of service level agreements or actual processing times. At this point, this step should already be firmly planned in the project schedule.

15 Optimising SOA Architectures

With advanced SOA projects, the resulting architectures can no longer be administered manually. Another phase in the SOA life cycle is reached – management and governance are required. For instance, information on how service failure affects the operational business is necessary. Furthermore, someone must verify who can change services and whether users and partners have the necessary access privileges. Also, the service quality and lifespan of individual components need to be recorded. In this phase at the latest an SOA librarian or "cybrarian" should support the project. This employee is familiar with, administers, and communicates all details of the existing systems and services. He or she raises the visibility of the SOA projects within the company through his or her commitment and helps define a set of rules and regulations.

16 The SOA Life Cycle can be controlled only by a Management Infrastructure

The necessary tools for this are included in SOA registries and repositories that offer more than one directory in accordance with the Universal Description, Discovery, and Integration (UDDI) specifications. The repository includes all SOA components, and stores processes, rules, service level agreements, availability, account policy, and other infrastructure details. It is only with the help of this management infrastructure that organisations are able to deal with the entire SOA life cycle.

17 Reorganise Process Steps

In the optimisation phase, companies take a look at the previously defined business goals and develop options for further improving the workflows. This is based on the results delivered by the measurement phase. In practice, process steps are rearranged here, further IT systems are added, and resources are tweaked for individual workflows.

An essential conclusion from customer SOA projects is that the question of management and governance of SOA infrastructures is gladly thrust aside: Whoever waits until the complexity can no longer be mastered is punished with additional project costs or delays.

18 Conclusion

As organisations continue to explore ways to implement an SOA, they need to consider a consistent vision and methodology throughout the process. This methodology should enable the organisation to:

- Assess and mitigate risk;
- Adopt best practices; and
- Link SOA implementation with key business initiatives.

Implementing SOA is about much more than adopting the latest technology or following the latest IT trends. It's about investing in the IT infrastructure to leverage existing IT assets in order to increase their reusability, thereby:

- Reducing the expense of integrating applications that are vital to each organisation's ability to remain competitive; and
- Providing a better level of service to internal and external customers.