

SOA Migration – Approaches and Experience

Rainer Gimmich

IBM Software Group

SOA Advanced Technology / Enterprise Integration Solutions

Wilhelm-Fay-Str. 30-34, D-65936 Frankfurt

gimmich@de.ibm.com

Summary

Due to their economic and technological benefits, Service Oriented Architectures (SOA) are valued more and more by companies in all industries and sizes. This paper explains SOA briefly and presents approaches for migrating to an SOA in 'real life'. We will discuss how SOA design methods and proven software reengineering technology can be combined in order to support a company's SOA adoption roadmap.

1. SOA

Service Oriented Architecture (SOA) is a business-centric IT architectural approach that supports integrating business as linked, repeatable business tasks, or **services**. Web Services provide a standardized, cost-effective implementation of such services (but this is not mandatory for an SOA). The services that a company wants to expose (internally or to the market) must be commonly understood by the business and the IT departments (see central 'layer' in Figure 1).

Then the business requirements can be implemented with a view to services, which also form major elements of the **business processes** to support: The services are invoked where they are required in the process; they can also be accessed directly from consumers (e.g. via a portal).

Apart from the processes there are other concepts that are important on the business side, for example the business competencies and their use, the business drivers and needs, the business performance context, etc. All of these provide valuable input for service modeling.

The **service model** includes identification of service candidates, service exposure decisions, service specification and realization decisions. The service specification includes service component specifications and flow specifications (both of services and components).

This leads us to the **service components** layer, where the Enterprise Components pattern has been applied successfully for grouping functionality. The service components usually rely on **operational systems** for their implementation (e.g. custom applications or software packages, 'legacy' in a positive sense).

This is a major feature of an SOA: the architecture is achieved in an evolutionary way, evaluating and migrating the existing landscape (or particular business areas) into an SOA in a stepwise manner. Each step can leverage its own benefits (higher reuse, faster time-to-market, lower development / maintenance effort, etc.).

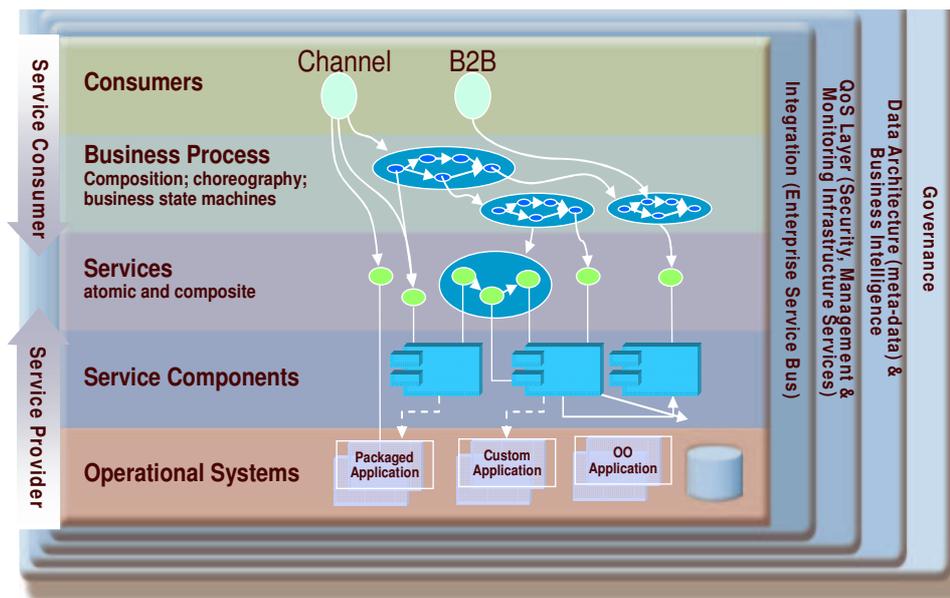


Figure 1: SOA conceptual layers

2. SOA Migration

Migration in IT means the move to a new technical environment, mostly for business reasons and for fulfilling (new) non-functional requirements. A software system is typically migrated in one or more aspects: to a different data management, communication, transaction, programming, or user interface environment (according to [Sneed et al. 2005]).

In addition, [Gimmich/Winter 2005] point out 'migration types' (dimensions) that can also apply in combination: migration of architecture, development environment, system software and hardware. The authors have sketched a sample SOA migration project, which

authors have sketched a sample SOA migration project, which

is – of course – an architecture migration but additionally entails introducing new tools into the development environment and new products for service monitoring, but no hardware changes.

[Channabasavaiah et al. 2004] show that SOA migration can help solve some of the application software problems we still face today: complexity, redundant and non-reusable programming, and multiple (point-to-point) interfaces. From their experience, an architecture migration into SOA is based on the following key requirements:

- **Leverage existing assets:** Existing systems (or assets from them) need to be included in the targeted SOA. They provide benefits such as suitability, efficiency, and stability. Often, this entails transforming existing assets but also, tactically, encapsulating and integrating existing assets is desirable.
- **Support all required types of integration:** Integration on user interaction (e.g. portals) level, application connectivity, information integration, process integration; potentially supported by design for integration and new component models such as SCA [SCA 2005].
- **Allow for incremental implementation and migration of assets:** Reduced migration complexity, plus the capability to produce incremental return-on-investment.

3. Project Examples

Let us consider two projects which materialize different approaches to SOA requested by their respective top-level management. Project 1 (in the finance industry) follows a top-down approach to service-modeling and selective implementation of services. Project 2 (in manufacturing) focuses on a bottom-up analysis of existing systems and the breadth of service delivery. Both projects make use of the same service modeling method, tailored to specific needs.

4. Service Modeling Support for Migration

Industrial-strength methods to model services are used by combined business and IT teams and provide important quality aspects in SOA implementation [Gimnich 2006]. Our method combines a generally top-down driven SOA design with bottom-up elements (application portfolio assessment and program understanding techniques).

In this approach, the business domain for SOA migration is analyzed using Component Business Modeling (CBM) or other methods for defining the business competencies in non-overlapping, largely stand-alone Business Components, while documenting their support to the business processes.

These business components provide a good starting point for designing services: each component has a set of consumed and offered services that are provided by other business component. Now the processes that use these business services can be decomposed, and the service candidates are refined. A second technique (goal-service modeling) provides further top-down and also middle-out support: looking at the business goals, sub-goals and key performance indicators to be achieved by the intended services. A third technique pro-

duces service candidates by **analyzing existing assets** (e.g. programs, APIs, transactions) already used in the domain. These three techniques are combined and complete the service identification. After this, service exposure decisions are made, and the resulting services then go into the specification and **realization decisions** phases. The latter phase includes analyses on whether a particular service should be based on existing assets – potentially by wrapping and **integrating** a legacy function or by **transforming** legacy –, or purchased on the market, or custom-developed.

The SOA design method used is called SOMA (Service Oriented Modeling and Architecture) [Arsanjani 2004].

5. Migration Planning and SOA Governance

Within each company, a specific SOA Roadmap needs to be defined, including all sub-projects to be performed in the SOA transition. A governance framework is established for the roles & responsibilities (e.g. design authority) involved in the transition, including the various migration tasks.

Wile Project 2 tends to have a stronger starting point for the technical migration tasks than Project 1, the business goals and the organizational aspects of the migration must also be met. So the method use is extended for business alignment and completeness ‘checks’ in the overall approach.

6. SOA Migration Tooling

The above-mentioned methods, in particular the CBM and SOMA approaches have to be supported as widely as possible in the development environment. One example implementation is the SOA Integration Framework (SOA-IF) used by IBM in a number of projects.

For legacy inclusion, support is required at least for analyzing existing assets (e.g. by interfacing the WebSphere Studio Asset Analyzer). Support in the legacy transformation decisions and actual transformation can be achieved using dedicated tools (e.g. Asset Transformation Workbench).

References

- [Arsanjani 2004] A. Arsanjani: Service-Oriented Modeling and Architecture. <http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/>, November 2004.
- [Channabasavaiah et al. 2004] K. Channabasavaiah, K. Holley, E. M. Tuggle: Migrating to a Service-Oriented Architecture. White paper, G224-7298, IBM, 2004.
- [Gimnich 2006] R. Gimnich: Quality Management Aspects in SOA Building and Operating. SQM 2006, <http://www.sqs-conferences.com/>, May 2006.
- [Gimnich/Winter 2005] R. Gimnich, A. Winter: Workflows der Software-Migration. WSR 2005, Softwaretechnik-Trends 25:2, May 2005. Presentation: <http://www.uni-koblenz.de/sre/Conferences/WSR/Wsr2005/Programm/gimnichwinter.pdf>
- [SCA 2005] Service Component Architecture. White paper V0.9, BEA/IBM/Interface21/IONA/Oracle/SAP/Siebel/Sybase, November 2005, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-sca/SCA_White_Paper1_09.pdf
- [Sneed et al. 2005] H. Sneed, M. Hasitschka, M.-T. Teichmann: Software-Produktmanagement. dpunkt, Heidelberg, 2005.