Analysis and Design Techniques for Service-Oriented Development and Integration

Olaf Zimmermann, Niklas Schlimm, Günter Waller, Marc Pestel

IBM Deutschland
Pascalstrasse 100
Stuttgart, Germany
{ozimmer, schlimmn, pestel, gwaller}@de.ibm.com

Abstract: Service-Oriented Architectures (SOAs) have been established as an IT strategy to support the on demand goal of business agility. Web services standards and their implementations are key enablement technologies for SOA which are maturing rapidly. There is a growing body of successful implementations of these technologies. However, experience of solving the wider business and architectural issues involved in designing a high-quality SOA for a particular enterprise still stands at an early stage. In this paper, we motivate the need for service modeling methodologies as means of tackling the external design of a business-focused SOA, identify some of the available candidate assets, and discuss how existing artefacts such as UML analysis diagrams can be leveraged for service modeling.

1 Motivation and Rationale for SOA

In a 2004 interview for Info World, Grady Booch stated that "the fundamentals of engineering like good abstractions, good separation of concerns never go out of style", but he also pointed out that "there are real opportunities to raise the level of abstraction again" [Bo04]. We believe that when designing large-scale enterprise applications, the abstraction level has to be raised up to the *business domains* a company deals with, taking the entire enterprise IT landscape into account. Business-aligned *software services* organized into an enterprise-scale SOA reside on such an increased level of abstraction.

While the SOA approach strongly reinforces well-established, general software architecture principles such as *information hiding, modularization*, and *separation of concerns*, it adds several themes pertaining to Enterprise Application Development and Integration (EADI). *Service composition* (also called *choreography* or *orchestration*), *service repositories* (broker role), and the *Enterprise Service Bus* (ESB) middleware pattern, are some of the major ones the industry has already identified. It is worth noting that SOA does not equal Web services – SOA meets the definition of an *architectural style* from [BCK98], and Web services are a highly attractive (but not the only) implementation alternative for SOA. The values provided by SOA include an increase of flexibility, agility, and better responsiveness to constantly changing business environments.

For example, in an SOA, service provider implementations can be replaced without impact on the service consumer – service interfaces declarations and ESB messaging decouple provider and consumer, and implementation details are encapsulated on the provider side. Process flows can be reassembled without coding effort on the atomic service layer – flow composition is isolated from computations and data management [Ar04].

In anticipation of the discovery of new business opportunities or threats, SOA as an architectural style aims to provide modular enterprise business solutions that can extend or change *on demand*. SOA solutions are composed of reusable services, with well-defined, open and published interfaces. Hence, SOA provides a mechanism for integrating existing legacy applications regardless of their platform or language.

2 From Business Models to SOAs via Service Modeling

At this point, we have identified the input and output of the service modeling process – but how about the steps between? A prescriptive modeling algorithm or at least detailed guidance is required that helps answer the following question:

How do we derive "good" service abstractions from high-level business requirements and business process models?

As a corollary, additional questions arise: what are good services, for example, what is the right service granularity, and what does business alignment mean from a modeling standpoint? How can an existing IT landscape be transformed into a services ecosystem?

When trying to answer these questions, a non-trivial EADI project often has to start from only vaguely articulated requirements, documented as high-level business process and/or use case models created by business analysts or consultants. In many cases, these models are defined only informally or semi-formally. However, formal service descriptions have to be defined eventually, as well as one or more realizations for each of them running in some IT infrastructure such as an application server or transaction monitor.

From a modeling standpoint, the resulting challenge is how well-designed, meaningful service abstractions can be characterized and constructed systematically:

- How are services in a SOA identified and described?
- What is the process for developing a SOA and services?
- How are business processes realized in terms of a SOA and services?
- Which development approaches are relevant to a SOA and service assets?
- How can legacy systems and packaged applications be adapted as services?

The related issues, which we jointly refer to as *service modeling* or *service-oriented analysis and design* [ZKG04], [Za05], currently are among the most frequently discussed ones in the industry and academia; we have not participated in a single SOA effort yet in which such service modeling aspects have not been a major issue, giving fuel for numerous debates. Elements from several methodologies and techniques served us well when encountering these issues on projects [Zi04], [Zi05].

3 Candidate Assets for Service Modeling

3.1 Service-Oriented Modeling and Architecture (SOMA)

SOMA [Ar04] is an IBM offering that defines the three service modeling steps *identification*, *specification*, and *realization*. These steps consist of several sub-steps prescribing several artifacts to be delivered and recommending appropriate techniques (Figure 1):

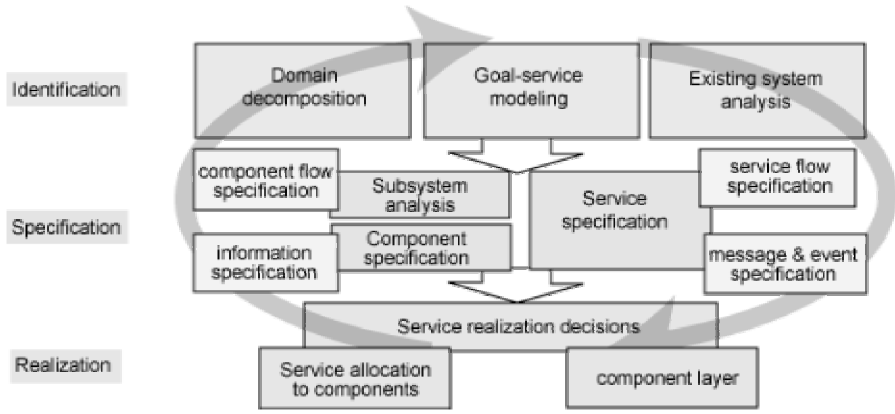

Figure 1: Services Identification, Specification, and Realization with SOMA (source: [Ar04])

SOMA identification can start both from business models via *domain decomposition*, which includes functional area analysis and process decomposition, and from existing systems. An additional *goal-service modeling* technique ties business goals, for example expressed as Key Performance Indicators (KPIs) to the identified service abstractions, facilitating runtime monitoring of business goals (a key business performance and service management issue).

The SOMA steps are performed in an iterative and incremental fashion. During service specification, the artifacts comprising an SOA are formally defined, for instance composite and atomic services, as well as components implementing them along with their interfaces. Collectively, these specifications form the *service model*, a key SOMA deliverable that covers service invocation syntax and semantics, as well as operational and other cross-cutting concerns such as service ownership, dependencies, versioning, and governance issues. Realization of services and components is business-as-usual from an application architect standpoint, at least to a large extent; well-established tools and techniques such as patterns, for example IBM Patterns for e-business, can be used.

3.2 Rational Unified Process (RUP) Extensions and Other Contributions

At present, many SOA extensions to general-purpose software development processes such as RUP are defined. For example, there is a RUP-SE configuration, as well as several proposals for Unified Modeling Language (UML) SOA Profiles [Jo05].

The CBDI Forum has recently published a proposal for a SOA methodology as well [Sp05]. An additional option would be to leverage Web services specifications such as Web Services Description Language (WSDL) and WS-Policy as core service model.

It is worth noting that best-of-breed or pick-and-choose adoption strategies are perfectly valid in our opinion – service-oriented analysis and design should be viewed as a refinement and enhancement of existing general-purpose methodologies and techniques for the particular problem domain of crafting SOAs of quality, for example Business Process Modeling (BPM), Object-Oriented Analysis and Design (OOAD), and Enterprise Architecture (EA) frameworks. Consequently, a custom adoption of several or all of the above mentioned assets is likely to be chosen on projects, depending on factors such as project scope, existing modeling artefacts, and methods and techniques already in use.

According to our experience, such a custom mix of BPM, OOAD, and EA covers large parts, but not all, of what is required for service-oriented analysis and design, which reemphasizes the need for methodologies like SOMA and complementary techniques such as a UML-centric service identification approach, one of which we outline next.

4 Identifying Candidate Services in UML Analysis Models

One of the key aspects of SOA modeling is a layering of services [Ar04]. For instance, the objective of the SOMA identification step is to establish candidate services residing on these layers. Several complementary techniques can be employed to accomplish this.

4.1 Service Types on the Business Level

Different viewpoints on services exist on the business and on the technical level. On the business level of abstraction, a *SOA service* is a business feature that is made available to customers, business partners or other interested parties of the organisation. This view of a service is completely decoupled from any technical aspects.

Based on the *value chain model* [Po85] and the structure of *business information systems* [FS01], [Si99], different types of services can be identified. The various types of services differ with respect to the value they provide to the business. For instance, services offered on a *value chain level* provide higher value then services on the *business process level*. Value chain level services typically are coarser grained than services offered by lower level business processes. Coarse grained services tend to have a rather low reuse potential, which means that reuse potential and business value correlate negatively.

4.2 A UML Meta-Model to Identify Service Types

Many businesses describe their business processes using UML [BRJ99] as modelling language. Today it is safe to say that UML has become an industry standard notation for software specification. Consequently, there is a need to describe precisely the procedure on how to identify candidate services within a given UML analysis model.

UML analysis models must adhere to a specific meta-model so that SOA services of different types can be identified. If they are in line with this meta-model, it is possible to derive the *service model* from the *business process model* and the *use case model*. Figure 2 illustrates these relationships and the central role of the *integrated meta-model*:

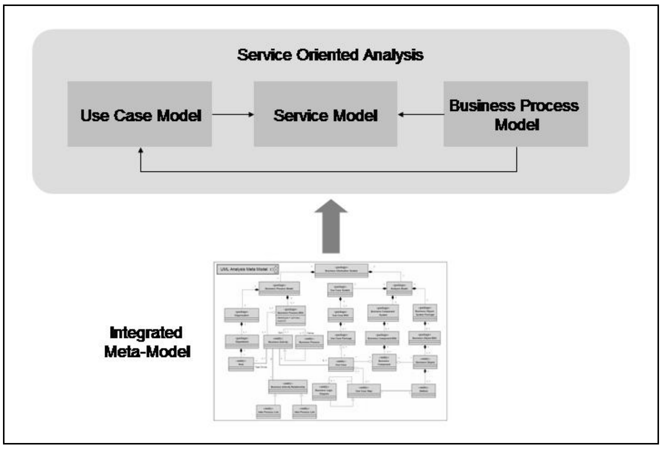

Figure 2: Role of Meta-Model in Service Oriented Analysis

Service-oriented analysis is the process that models business domains with the objective to create a SOA. On this level, the integrated meta-model describes the modeling rules within the three model views use case model, business process model and service model. The first two are UML-based models whereas the service model is a new artefact within SOMA (as outlined in Section 3). It is important that the meta-model integrates these different views, as concepts from the business process model and use case model have to be mapped against concepts in the service model.

This integrated analysis-level meta-model can be used in EADI engagements during the SOMA identification step. For example, it can facilitate top-down domain decomposition. It can also be used to derive service candidates from any existing UML analysis model, for example during SOMA existing system analysis. In the latter case, this implies that the existing UML model must meet specific requirements.

5 Lessons Learned and Conclusions

In summary, successful service modeling is as not as easy as it might appear at first glance; much more than simple drill-down from business-level process flow to IT realization is required, and many SOA-specific architectural decisions have to be taken.

Almost all but the most trivial cases require a meet-in-the-middle modeling approach as opposed to a top-down process, because existing system reality constrains modeling choices, for example software packages used for Enterprise Resource Planning (ERP).

Related tools and techniques are under construction at present, and corresponding best practices are emerging. Methodologies such as SOMA and SOA-specific patterns can and should be leveraged to ensure repeatability, and support quality assurance and risk mitigation strategies. Due to space constraints, we could only give a brief overview of one SOA-specific modeling methodology and a related analysis technique; we refer the reader to [Ar04] for more information.

Service modelling activities always have to be adapted to client and project environment; combining elements from several methodologies and techniques is a valid option. UML models and other architectural artefacts such as system context and component interaction diagrams can play a key role during analysis and early design. As an example, in this paper we outlined how a candidate service model can be derived from existing UML analysis artefacts such as use case and business process models.

References

[Ar04] Arsanjani, A.: Service-Oriented Modelling and Architecture (SOMA), IBM developer-Works 2004, http://www.ibm.com/developerworks/webservices/library/ws-soa-design1

[BCK98] Bass, L.; Clements, P.; Kazman, R.: Software Architecture in Practice, Addison Wesley, 1998.

[Bo04] Booch, G.: InfoWorld interview 2004, http://www.infoworld.com/article/04/02/02/HNboochint_3.html

[BRJ99] Booch, G.; Rumbaugh, J.; Jacobson, I.: The Unified Modeling Language User Guide, 1999

[FS01] Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik, 2001

[Jo05] Johnston, S.: UML 2.0 Profile for Software Services, IBM developerWorks 2005, http://www.ibm.com/developerworks/rational/library/05/419_soa

[Po85] Porter, M.: Competitive Advantage, 1985

[Si99] Sinz, E. J.: Konstruktion von Informationssystemen. Bamberger Beiträge zur Wirtschaftsinformatik Nr. 53, Oktober 1999

[Sp05] Sprott, D.: On SOA Methodology, Editorial March 2005 CBDI Journal, http://www.cbdiforum.com/report_summary.php3?page=/secure/interact/2005-03/editorial.php&area=bronze

[Za05] Zacharias, R.: Serviceorientierung: Der OO-König ist tot, es lebe der SOA-König!, OBJEKTSpektrum 02/2005 (in German)

[Zi04] Zimmermann, O.; Milinski, M.; Craes, M.; Oellermann, F.: Second Generation Web Services-Oriented Architecture in Production in the Finance Industry, OOPSLA Conference Companion, 2004

[Zi05] Zimmermann, O.; Doubrovski, V.; Grundler, J.; Hogg, K.: Service-Oriented Architecture and Business Process Choreography in an Order Management Scenario, OOPSLA Conference Companion, 2005

[ZKG04] Zimmermann, O.; Krogdahl, P.; Gee, C.: Elements of Service-Oriented Analysis and Design, IBM developerWorks 2004, http://www.ibm.com/developerworks/webservices/library/ws-soad1/index.html