# A Survey of Service Oriented Development Methodologies

Ervin Ramollari[1], Dimitris Dranidis[1], and Anthony J. H. Simons[2]

[1] South East European Research Centre (SEERC)
17 Mitropoleos Str., 54624 Thessaloniki, Greece
{erramollari, ddranidis}@seerc.org
[2] Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street,
Sheffield, S1 4DP, UK
a.simons@dcs.shef.ac.uk

**Abstract.** Service orientation is a new software engineering paradigm that introduces opportunities as well as challenges. Although existing processes and practices can be reused for service oriented development, novel techniques are required to address unique SOA requirements. Work in this area is quite active and only recently is producing some initial results. The aim of this paper is to present a state-of-the-art survey on current service oriented development approaches. The characteristics that distinguish between these approaches are discussed and a number of actual methodologies that have emerged or are still emerging are described and compared.

**Key words:** SOA, service oriented software engineering, methodologies, survey

## 1 Introduction

Service-Oriented Computing represents a paradigm shift in software engineering, where the key abstraction is that of *services*, utilized to support rapid and low-cost application development through service composition. While technology and standards, such as Web services, are important to achieve SOA, it has been widely recognized that they are not sufficient on their own. Just by applying a Web service layer on top of legacy applications or components does not guarantee true SOA properties, such as business alignment, flexibility, loose coupling, and reusability. Instead, a systematic and comprehensive approach is of critical importance, taking into account the business requirements and following recommended practices. As Gartner [1] had predicted, "by 2007, 70 percent of SOA and Web services engagements will require a cohesive, end-to-end service delivery methodology and tool set".

A number of preliminary methodologies have emerged to address the huge demand for process guidance and proven best practices in SOA projects. However, a survey on these methodologies and an analysis of their properties is currently lacking. Related work mainly treats service-oriented methodologies from

a general point of view without referring to specific proposed ones. Arsanjani from IBM [2] broadly classifies SOA approaches under six categories: *business process driven*, *tool-based MDA*, *wrap legacy*, *componentize legacy*, *data-driven*, and *message-driven*. Papazoglou et al [3] provide a research roadmap, where among other things, they briefly explore the state of the art and some grand challenges in service oriented engineering. Zimmermann et al [4] discuss about analysis and design techniques for service-oriented development and integration, with IBM SOMA method as an example.

This paper goes into more depth by surveying actual approaches and methodologies. The characteristics and criteria that are used for comparison are discussed first, and then the actual methodologies are presented and compared using these characteristics.

## 2 Characteristics of SOA Development Methodologies

Below we present the criteria that we use to evaluate and compare SOA development approaches:

**Delivery strategy:** There exist three common strategies in delivering a SOA, depending on the amount of front-end analysis of the business domain and the treatment of existing legacy systems [12]. The *top-down* strategy is closely tied to an organizations' existing business logic, from which required services are derived. The *bottom-up* strategy is the opposite in that it focuses on legacy systems, and Web services are built on an as-needed basis. The *meet-in-the-middle* (agile) strategy finds a balance between incorporating service-oriented design principles into business analysis environments without having to wait before integrating Web services technologies into technical environments [12].

**Lifecycle coverage:** Some proposed approaches aim to support the full SOA lifecycle, including planning, analysis and design, construction, testing, deployment, and governance activities, while others limit their scope to a subset of these phases, such as analysis and design.

**Degree of prescription:** SOA methodologies range from the most prescriptive ones that specify phases, disciplines, tasks, and deliverables for each of them, while others provide less detail, by purpose or not, leaving room for more flexibility and tailoring of the approach depending on the project context.

**Availability:** A number of methodologies proposed by industry players such as IBM, Sun, Microsoft, and others, are proprietary and the detailed specifications are not openly available. In contrast to open methodologies whose documentation is available to the interested public, for the proprietary methodologies it is difficult to fully analyze their capabilities and to make comparisons.

**Process agility:** A number of methodologies suggest an agile approach to Service Oriented development in order to address risks and add flexibility to change. Yet, some others follow a more rigid approach in the process lifecycle, or do not address the issue of agility at all.

**Adoption of existing processes/techniques/notation:** A large number of SOA methodologies propose reusing proven existing processes like XP and RUP,

and techniques like OOAD, CBD, and BPM, seeing service-oriented development as an evolutionary rather than revolutionary step in software engineering. Also standardized notations, such as UML and BPMN, are being adopted to visually model various artefacts.

**Industrial application:** It is important that a methodology be validated in proof-of-concept case studies to show that it has practical applicability and to refine it based on feedback from the case studies. Unfortunately, most of the existing SOA methodologies are at an early stage and have not been applied yet in industrial projects.

**Supported role(s):** A service-oriented methodology may support the provider view, the consumer view, or both the provider and consumer views in an integrated framework. In the consumer's view, development is declarative and business process oriented through service composition, while in the provider's view it is programmatic and component oriented.

## 3 Analysis of Existing Methodologies

**IBM Service-Oriented Analysis and Design (SOAD) [5]:** SOAD proposes elements that should be part of a service-oriented analysis and design methodology, hence it is an abstract famework rather than a holistic methodology. SOAD builds upon existing, proven techniques, such as OOAD, CBD, and BPM. It also introduces SOA-specific techniques, such as service conceptualisation, service categorization and aggregation, policies and aspects, meet-in-the-middle process, semantic brokering, and service harvesting.

**IBM Service Oriented Modeling and Architecture (SOMA) [6]** SOMA is a full-blown modeling methodology by IBM consisting of three steps: *identification*, *specification*, and *realization* of services, flows (business processes), and components realizing services. The process is highly iterative and incremental. However, because SOMA is proprietary to IBM, its full specification is not available. It has been recently announced that the Rational Unified Process has been combined with SOMA to result in what is called *IBM RUP for SOMA* [15].

**SOA Repeatable Quality (RQ) [7]:** SOA RQ is a proprietary methodology by Sun Microsystems that is based on a RUP-like iterative and incremental process consisting of five phases: inception, elaboration, construction, transition, and conception. UML compliant artefacts are used for documenting various deliverables of these phases.

**CBDI-SAE Process [8]:** The CBDI Forum is currently developing a SOA methodology as part of its CBDI-SAE SOA Reference Framework (RF). The four key discipline areas of the process are: consume, provide, manage, and enable. Each area groups similar disciplines that are further broken down to process units and then to tasks. This methodology aims business-IT integration through top-down analysis of business requirements as well as bottom-up legacy system integration. The CBDI-SAE process aims to cover the whole SOA lifecycle, including deployment, monitoring, and governance activities.

**Service Oriented Architecture Framework (SOAF) [9]:** SOAF consists of five main phases: information elicitation, service identification, service definition, service realization, and roadmap and planning. It is concurrently based on two types of modeling activities: "To-be" modeling, which is the top-down business oriented approach describing the required business processes, and "As-is" modeling, which is the bottom-up approach describing current business processes as they are shaped by the existing applications.

**Service Oriented Unified Process (SOUP) [10]:** As the name suggests, this approach by K. Mittal is primarily based on the Rational Unified Process. Its lifecycle consists of six phases: incept, define, design, construct, deploy, and support. However, SOUP lacks detailed documentation and leaves room for adaptation. It is used in two slightly different variations: one adopting RUP for initial SOA projects and the other adopting a mix of RUP and XP for the maintenance of existing SOA rollouts.

**Methodology by [11]:** In their paper, Papazoglou et al examine a service development methodology from the point of view of both providers and consumers, which attempts to cover the full SOA lifecycle. It is partly based on well-established development models, such as the RUP, CBD, and BPM. The methodology utilizes an iterative and incremental process that comprises one preparatory and eight distinct main phases.

**Thomas Erl's [12]:** The service oriented analysis and design methodology documented in Thomas Erl's book [16] is considered the first vendor-agnostic one to be published . This methodology is a step by step guide through the two main phases: analysis and design. The activities in the analysis phase take a top-down business view where service candidates are identified. These serve as input for the next phase, service oriented design, where the service candidates are specified in detail and later realized as Web services.

**BPMN to BPEL [13]:** In this approach the business process is expressed in an abstract model (Business Process Modeling Notation or BPMN) and according to transformation rules it is automatically mapped to an execution language (Business Process Execution Language or BPEL) that can be executed by a process engine. The authors in [13] coined the term *business process oriented programming* to refer to an evolutionary step in software engineering where programming power is given to the business analyst.

**Steve Jones' Service Architectures [14]:** The scope of this top-down methodology consists of the first steps in a project necessary to ensure that true SOA properties are satisfied in the final delivery. It is technology agnostic and takes a top-down business view reaching up to the point of service candidate discovery (i.e. identification). The methodology adopts a broadly four-step process (What, Who, Why, and How), of which the first three are covered in preparation for the fourth step.

Comparison of the listed methodologies according to the identified characteristics is summarized in the table below.

| | IBM SOAD | IBM SOMA | SOA RQ | CBDI-SAE | SOAF |
|---|---|---|---|---|---|
| **Delivery strategy** | M | M | M | M | M |
| **Lifecycle coverage** | A&D | A&D | complete | complete | A&D and planning next phases |
| **Prescriptive** | 1 | 4 | 3 | 4 | 3 |
| **Proprietary** | yes | yes | yes | no | no |
| **Agile** | n/a | 3 | 4 | 2 | 2 |
| **Existing process** | no | RUP (recently) | RUP | ? | no |
| **Existing techniques** | OOAD, BPM | ? | ? | ? | no |
| **UML** | yes | ? | yes | ? | ? |
| **Applied in industry** | yes | extensively | extensively | not yet | a case study |
| **Consumer view** | yes | yes | yes | yes | yes |
| **Provider view** | yes | yes | yes | yes | yes |

| | SOUP | Papaz. | Erl's | BPMN to BPEL | Jones' SA |
|---|---|---|---|---|---|
| **Delivery strategy** | M | M | T | T | T |
| **Lifecycle coverage** | complete | complete | A&D | A& D and Impl. | Initial planning |
| **Prescriptive** | 1 | 2 | 4 | 2 | 1 |
| **Proprietary** | no | no | no | no | no |
| **Agile** | 5 | 3 | 1 | n/a | n/a |
| **Existing process** | RUP, XP | RUP | no | no | no |
| **Existing techniques** | no | CBD, BPM | BPM | BPM | no |
| **UML** | no | no | no | no | no |
| **Applied in industry** | not yet | not yet | not yet | not yet | not yet |
| **Consumer view** | yes | yes | yes | yes | yes |
| **Provider view** | yes | yes | yes | no | no |

**Table 1.** Comparison of SOA development methodologies. (A relative quantitative scale 1-5 is used for some criteria. Also, M = Meet-in-the-Middle, T = Top-Down, B = Bottom-Up, and ? = No Data)

## 4 Conclusions

In this paper we presented a state of the art survey of the current service oriented engineering approaches and methodologies. One interesting point is that current SOA methodologies build upon existing, proven techniques, such as OOAD, EA, and BPM. Also, agile processes like XP and RUP are being employed successfully in SOA projects. However, the service paradigm introduces unique requirements that should be addressed by innovative techniques. Another interesting point is that most of the surveyed SOA methodologies propose the meet-in-the middle strategy, where both business requirements and existing legacy applications are taken into account to derive services. Although top-down analysis of the business domain produces services of high quality and long-term value, reality constraints require existing investment on IT infrastructure to be incorporated as well.

Generally, the service oriented development methodologies that have emerged are quite new and do not yet offer the required level of maturity. It is too early to determine whether any one of these methodologies is more appropriate than the others, or even to consider unifying some of them into a widely acceptable standard, as has been the case with the Rational Unified Process and the UML for object orientation. Therefore, we could say that this is a time of "methods war" for service oriented engineering that will eventually result in well-established and standardized methodologies.

## References


1. M. Cantara, "Common features of external service providers' SOA frameworks and offerings", Gartner, September 2005.
2. A. Arsanjani, "Toward a pattern language for Service-Oriented Architecture and Integration, Part 1: Build a service eco-system", IBM Corporation, available from http://www-128.ibm.com/developerworks/webservices/library/ws-soa-soi/, July 2005.
3. M. P. Papazoglou et al, "Roadmap of Service Oriented Computing", March 2006, available from http://infolab.uvt.nl/pub/papazogloump-2006-96.pdf.
4. O. Zimmermann et al, "Analysis and design techniques for Service-Oriented Development and Integration", available from http://www.perspectivesonwebservices.de/download/INF05-ServiceModelingv11.pdf.
5. O. Zimmermann et al, "Elements of Service-Oriented Analysis and Design", IBM Corporation, available from http://www-128.ibm.com/developerworks/library/wssoad1/, June 2004.
6. A. Arsanjani, "Service-oriented modeling and architecture", IBM Corporation, available from http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/, November 2004.
7. SUN Microsystems, "SOA RQ methodology - A pragmatic approach", available from http://www.sun.com/products/soa/soa_methodology.pdf.
8. P. Allen, "The service oriented process", in *CBDi Journal*, February 2007, http://www.cbdiforum.com/report_summary.php3?page=/secure/interact/2007-02/service_oriented_process.php&area=silver.
9. A. Erradi et al, "SOAF: An architectural framework for service definition and realization", in *Proceedings of the IEEE International Conference on Services Computing*, pp 151-158, Chicago, USA, September 2006.
10. K. Mittal, "Service Oriented Unified Process (SOUP)", available from http://www.kunalmittal.com/html/soup.shtml, 2006.
11. M. P. Papazoglou and W. J. van den Heuvel, "Service-oriented design and development methodology", *International Journal of Web Engineering and Technology (IJWET)*, 2006.
12. T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Upper Saddle River: Prentice Hall PTR, 2005.
13. C. Emig et al, "Development of SOA-based software systems - and evolutionary programming approach", in *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*, p 182, Guadeloupe, French Caribbean, February 2006.
14. S. Jones, "A Methodology for Service Architectures", Capgemini UK plc, available from http://www.oasis-open.org/committees/download.php/15071/A%20methodology%20for%20Service%20Architectures%201%202%204%20-%20OASIS%20Contribution.pdf, August 2005.
15. Ali Arsanjani, "IBM's SOA method: SOMA, Service-Oriented Modeling and Architecture", IBM Corporation, available from http://www-03.ibm.com/developerworks/blogs/page/AliArsanjani?entry=soma_service_oriented_modeling_and&ca=drs-bl, December 2006.
16. "SOA-Glossary", Cambridge Technology Enterprises, available from http://www.ctepl.com/soaterms.shtml.