An Overview of Software Engineering Approaches to Service Oriented Architectures in Various Fields

Artemios Kontogogos and Paris Avgeriou

University of Groningen

Department of Computer Science

P.O. Box 407, 9700 AK Groningen, The Netherlands

a.kontogogos@rug.nl , paris@cs.rug.nl

Abstract

For the last few years, a rise has been observed in research activity in Service Oriented Architectures, with applications in different sectors. Several new technologies have been introduced and even more are being currently researched and aimed to the future. In this paper we present and analyze some of the most influential approaches from a software engineer's point of view that belong either to the academic or to the industrial field. Despite their differences though, all of these approaches share a service oriented mentality, with the purpose of lessening the issues of clients and companies, students and teachers, citizens and government employees alike. Lastly, we discuss our findings from the comparison and present possible new research opportunities for the immediate future.

Keywords: Survey, Software Engineering, Service Oriented Architecture (SOA), Service-Centric Systems, Software as Service(SaS).

1. Introduction

Service Oriented Architectures is a relatively new branch of the Software Engineering field. The importance of SOA applications for the highly evolved and technology-based societies of today is apparent. The service centric approach has a number of benefits compared to other methodologies, such as the ability of services to function as self-contained and reusable units that can be located and executed on demand [28]. It is considered to be the new paradigm for developing flexible and dynamic software solutions by using loose coupling of services with diverse operating systems, programming languages and other components of various platforms. The integration of these interoperable services around a business process allows systems development in environments that have to continuously change and adapt to new circumstances. We hope that we will also be able to contribute to this effort with the SAS-LEG[1] project.

In order to achieve this kind of flexibility and customization to almost every situation and the ever-changing needs of the customers/users, a characteristic which is highly requested in all the fields that we analyze, we are obliged to improve and evolve the methodologies of the past such as Component Based Development (CBD) [26] and Object Orientation (OO) [10]. Moreover, we have to concentrate on discovering new ways and constructing new sets of tenets to abide by for developing software systems that will be compatible with the open world assumption and thus able to perform even in the most competitive real-life environments [5]. This survey is not exhaustive as can be seen from the number of the described approaches. An effort was made to include frameworks that are representative of the field and at the same time have a sufficient amount of information available. Lastly, we would like to clarify that currently there are a lot of disciplines that are related to SOA and use various names, such as Service-Centric Systems or Software as Service. In this paper we don't exclude these approaches, instead we study them and refer to all of them as Service Oriented Architectures. We approach this from a software engineer's point of view by including and using information such as lifecycle, processes, activities and workflows. Moreover the analysis is made with an emphasis on the academic characteristics of the selected SOA frameworks.

This paper is structured as follows. Firstly, in section 2 we reflect upon the various solutions proposed in the SOA area and present their characteristics. Thereafter we perform a comparison, refer to the benefits and drawbacks of each approach and their possible applications in different environments. Finally, we discuss the current state of af-

---

[1]SAS-LEG, Software As Service for the varying needs of Local e-Governments is a research project that proposes to use the Software as Service (SaS) principle in order to implement the law once and offer it as a customizable service to several municipalities (http://www.sas-leg.net).

IEEE computer society

fairs in the fields of our concern (such as e-Commerce, e-Learning, e-Government) and provide an estimation of the impact of these technologies on the events of the future.

2. Presentation of existing and emerging SOA methods and technologies

One of the projects concentrated on the service oriented methodology is Service Centric System Engineering (SeCSE)[2]. It is a European Union funded project that attempts to set the basis for the service centric approach. SeCSE tries to achieve this goal by creating the necessary tools and methods for the systems integrators and providers alike, while at the same time improving the cost efficiency of the development process [1]. There are some vital research results that apply to the entire SeCSE project and provide the necessary framework for the functionality of the more specified technologies [4]. One of these results is an effort to present a common conceptual model and a standardized language to describe the disciplines involved [16], which is essential for an effective communication of the researchers that participate in the project. Moreover, these results can be used outside the scope of the project with a certain amount of adaptation to the possible service oriented scenarios.

Other than that, the area of research was divided into four main active directions, which we are presenting next. The first sector of SeCSE is the service engineering direction which is concentrated on the specifications of the various services [22] and the testing methods used to ascertain their quality. The underlying issue here is that many of the established methods and tools used for the determination of the Quality of Service(QoS) property of a software system do not work in the case of services [14]. Therefore a new approach was proposed by the project in the form of test cases [11] and at the same time an effort took place in order to improve QoS description [18] and dependability [17]. In the service discovery branch of the project, the team proposed a supporting framework for the run-time service discovery [30] and also a search engine was developed with the ability to determine the best service for a high-level business model [1].The next sector is the sector of system engineering and is considered to be the centre of the entire project [1]. In this strand the focus was made on the QoS-Aware Service Composition, in order to solve the replanning problem with real time services and various solutions were presented [12] [15] [13]. Finally, the service delivery sector dealt with the monitoring and the correct delivery of services. Some of the research results of this field include Self-healing Services Compositions [8], Requirement Monitoring [24] [25],Smart Monitoring [7], Non
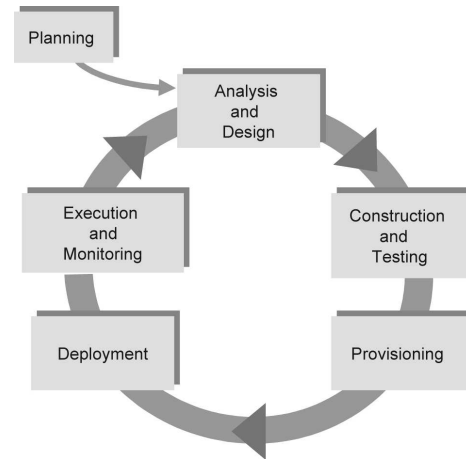

Figure 1. Phases of the service-oriented design and development methodology [27].

Intrusive [29] and Dynamic [9] Monitoring in various situations.

Another proposed by the academic community project was Service-Oriented Development In a Unified fraMework (SODIUM)[3], which involved international research, technological and industrial partners. It started in 2004 and finished in 2007. The project was focused on the unified discovery and composition of heterogeneous service types, by providing the necessary languages and software tools [2]. There are many heterogeneous types of services, such as web, grid and p2p services and the potential in combining these services was not being used fully. That was happening due to the inability of the existing service discovery mechanisms to handle heterogeneous service types, registries, networks and the various service description protocols. SODIUM proposes as a solution a set of models, languages (such as a Unified Service Query Language (USQL) which is an XML based language), middleware and other tools (such as the USQL search engine which is mainly used for querying) [31].

In their paper, Papazoglou et al [27] 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 (Rational Unified Process), CBD, and BPM (Business Process Management). The methodology utilizes an iterative and incremental process that comprises one preparatory and eight distinct main phases. These are planning, Analysis and Design (A&D), construction and testing, provisioning, deployment, execution and monitoring (Figure 1). The planning phase is used in order to organise all the stages of the methodology. The
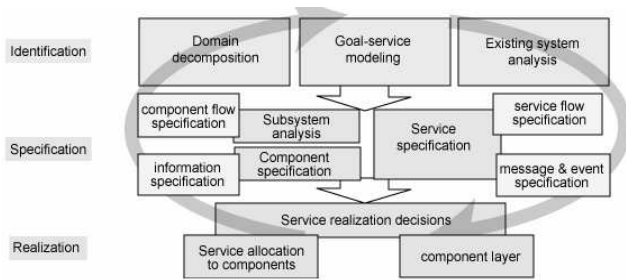
---

[2]http://www.secse-project.eu/

[3]http://www.atc.gr/sodium

Figure 2. The service-oriented modeling and architecture method [6].

analysis phase is based on a business case analysis which is used to decide the business processes that will be implemented. Service construction and testing includes coding the business processes that were selected and testing the services for functionality. The service-provisioning includes service rating and service billing. The last stage performs the deployment and the monitoring of the services. This approach is called Service Development Lifecycle Methodology (SDLM) and the iteration technique it employs allows the developers to improve their project in a manner of repeatable and predictable steps. Due to this multi step logic, the adaptability of the approach improves to a large degree.

Finally, the last academic related approach we are analyzing here is Service Oriented Architecture Framework (SOAF), which focuses around the business process with the goal to make the design and development of SOA easier [21]. The framework is a combination of five main phases: information elicitation, service identification, service definition, service realization, and roadmap and planning. The framework encompasses methods and techniques used to identify and model services and also to find services granularity[23].

From a different perspective we have a method which is extensively used in the industry, the Service-Oriented Modeling and Architecture (SOMA), which is an IBM Methodology for analysis and design [6]. It consists of three steps: identification, specification, and realization of services, business processes, and components realizing services. The process is highly iterative and incremental. A graphical representation of the process can be seen in Figure 2. The description of the steps of the process is below:

- Service Identification

  1. Domain decomposition, which consists of the decomposition of the business domain into its functional areas and subsystems, including its flow decomposition into processes and high-level business use cases.

  2. Existing system analysis, in this step existing systems are analyzed and selected as possible candidates for improving the cost of the implementation of service functionality that supports the business process.

  3. Goal-service modeling, used to validate and discover other services not selected by the previous service identification approaches.

- Service classification or categorization

  This activity is started when services have been identified. Classification helps determine the composition and the layering structure.

  1. Subsystem analysis, which takes the subsystems found above during domain decomposition and specifies the interdependencies and interactions between them.

  2. Component specification. In the next major activity, the details of the component that implement the services are specified.

- Service realization

  This step regulates the options for the software that implements a given service (selected,custom built or partly outsourced). Other realization decisions such as various business functionalities, security and monitoring of services also happen during this step.

  1. Service allocation is responsible for assigning services to the subsystems that have been identified so far.

Since the SOMA method was proposed, a significant amount of research was done in various fields that expanded and improved the original idea [32].

A specific method including various rules and structures for engineering an enterprise IT architecture according to SOA paradigm was proposed by G.Engels et al. [20]. Firstly, the framework is divided to business and IT aspect areas with the business area including the Modelling business services activity and the Business services entity which is connected to IT activities Designing domains, Designing components and Designing interfaces. These activities produce the analogous entities such as Domains, Components and Interfaces. Modelling business services consists of a number of steps:

- Identify top-level business services (such as plan, purchase, produce, sell and service).

- Identify service actions (such as the actions of the service sell: compose travel, book travel, and transact payment, which are also finer-grained services).

Table 1. Comparison of SOA methodologies (*next stages in planning).

Name	Description	Behaviour	Lifecycle	Detail	Adaptability	Industry
SECSE	multi-sided service specs	BPEL-like service composition	A&D	high	high	no
SODIUM	UML models with constraints	graphical composition language	full	medium	low	case study
SDLM	based on the business process	incremental development module	full	low	medium	no
SOAF	based on the business process	methodology framework	A&D*	medium	low	case study
SOMA	component based (partially on RUP,BPM)	business process flows	A&D	high	medium	many
G.Engels et al.	UML models with constraints	business process flows	A&D	medium	medium	yes
CBDI-SAE	process units	methodology framework	full	high	low	no

- Refine business services, which happens when certain conditions are met (e.g. when there are multiple service providers and business goals or when a suitable level of granularity is reached through the iteration of the two previous steps).

- Specify business services, where they are specified like use cases.

A development process that covers both the development of business architecture as well as the appropriate software architecture also has been proposed [19].

Another interesting industrial approach is being developed by the CBDI Forum. It is a SOA methodology that is a part of their CBDI-SAE SOA Reference Framework [3]. The four key discipline areas of the process are:

- Consume

  In this discipline, software solutions are created that are focused on the consumption of services and through that we gain better understanding of business requirements,processes and ultimately products.

- Provide

  In this sector, the identification and the design of services takes place, after that there is the planning and provisioning of services and finally their implementation. Of course, all of the steps are based on service specifications.

- Manage

  Here they discover the differences between the existing and the wanted SOA attributes and afterwards design the process that makes an effort to implement the missing attributes.

- Enable

  A platform is created that supports the implementation of the services and their run-time monitoring and even changing if it is required.

Each area groups similar disciplines that are further broken down to process units and then to tasks. This methodology aims at business-IT integration through top-down analysis of business requirements as well as bottom-up legacy system integration. The CBDI-SAE process aims to include SOA deployment, monitoring, and governance activities.

3. Comparative analysis of the Software Oriented Architectures

In this section we make an effort to present and compare the SOA we analyzed earlier. In order to support this goal we will use the below criteria:

- Service description, which is the way used by the approach to describe the relative services.

- Behaviour specification is the representation used by the approach to describe its behaviour.

- Lifecycle coverage, that shows what stages of the lifecycle methodology the approach covers [27].

- Detail of the approach is the degree of detail(e.g. stages,tasks,deliverables) provided by the approach.

- Adaptability of the approach is the ability of the approach to be flexible if any changes of the requirements or other parameters occur.

- Industrial applications show if there are any industrial case studies for the specific approach.

The above criteria were used in order to provide an image of the various characteristics of the chosen approaches from an academic point of view. For this reason, approaches without the needed amount of academic information background (such as the ones provided by suppliers like SAP and Oracle) were considered out of scope for this paper. This is also true for the industrial comparison criteria such as the cost of the implementation, the level of readiness in the organization and the suppliers that could support the research result.

We would like to make some observations about the methodologies we described in the previous section. Two of them propose language structures in order to represent and describe various sectors of their work. SECSE offers a general conseptual model that describes actors, activities, entities and the interactions between them [16]. It's goal is to provide some common ground for the rest of the project and possibly for others projects as well since because of the model's implementation it can be adapted rather easily. In order to portray the model UML based diagrams are utilized

but the authors also specify that any ontology-based language could be used instead. SODIUM on the other hand proposes three languages [2]. The first one is the Visual Service Composition Language (VSCL), which is a UML based graphical composition language with a specific purpose of defining service compositions containing heterogeneous services. VSCL also has concepts for defining QoS requirements that are used as criteria for service discovery, sorting and selection and are considered to be essential for the service discovery process which is supported by the Unified Service Query Engine. The Engine is based on the second proposed language, which is the USQL that we refered to in section 2. Lastly, there is the Unified Service Composition Language (USCL) which is necessary for the execution of heterogeneous service compositions. SECSE does not have such capabilities but its scope is different and it is focused on more general aspects of defining service-related consepts and at the same time it is able to offer things outside of SODIUM's reach (such as service publication, delivery and monitoring).

Comparison of the approaches using criteria we described in the beginning of the section can be seen in Table 1. Each approach portrayed here is unique and offers different characteristics that could benefit the fields that require high software agility and continuous improvement. The SDLM and SOAF approaches with their business procedures, incremental module and methodology framework could be easily implemented for the E-Commerce field. Moreover, the same applies for SOMA, CBDI-SAE and G. Engels method which is apparent, since for the first one there is a large number of cases when it was used in the industry and the second and third ones also have an industrial approach (G.Engels method also has been used in the industry). The SODIUM model could support any environment using heterogeneous services and SECSE with its high adaptability and detail possibly could be used in all the fields of our interest and possibly some others.

4. Conclusions and Outlook

In this paper we presented an overview of the current service oriented engineering approaches and techniques. We believe they can be highly useful and effective for fields such as e-Government, e-Commerce and e-Learning by providing the necessary structures and tools to support these complex and demanding environments. Based upon these approaches, advanced software systems can be created that will offer the right degree of automation and control at the same time. The need for these systems will only rise as the size and the requirements for services in the fields we mentioned increase. Therefore it is essential to prepare for such a likely event.

We can observe that the current SOA implementations

are a mix and adaptation of the earlier approaches such as CBD, OO, BPM and processes like RUP. Of course every approach we examined, chooses its own path to achieving the goals set before it, through the proposed innovational techniques and methodologies. While that shows that there are still a lot of possibilities for the growth of the SOA research field , at the same time it identifies a lingering issue. The approaches do not seem to share some common ground that could be used in the future to create a unified approach, as it happened earlier with RUP or UML. In order to support such a unification, firstly there should be created a satisfying number of widely excepted standards, models, patterns and language terms which would enable a more streamlined communication between the research groups. That would lead to more opportunities for joint projects. Thus, our opinion is that the maturity and effectiveness of the approaches we examined could vastly improve in the next few years. Moreover, as we can see from Table 1, some of the approaches still lack industrial application. Thus we believe that there is room for improvement and so a more active development of specific real-life projects or case studies and an improved tool support would greatly enrich the SOA domain.

Acknowledgements

References

[1] Putting services at the heart of tomorrows software. http://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/st-ds/secse_en.pdf. February 2009.

[2] S. Topouzidou. SODIUM, Service-Oriented Development In a Unified framework, Final report ISTFP 6-004559. http://www.atc.gr/sodium/upload/publications/SODIUM_Final_Report.pdf. Febrouary 2009.

[3] SP. Allen, The service oriented process, in CBDi Journal, February 2007. http://www.cbdiforum.com/secure/interact/2007-02/service_oriented_process.php February 2009.

[4] The SeCSe team, Towards Service-centric System Engineering. In proceedings of eChallenges e-2005, 19 - 21 October 2005, Ljubljana, Slovenia. http://home.dei.polimi.it/baresi/papers/eChallenges05.pdf. February 2009.

[5] W.J. van den Heuvel, O. Zimmermann, F. Leymann, P. Lago, I. Schieferdecker, U. Zdun, and P. Avgeriou, Software Service Engineering: Tenets and Challenges, proceedings of the Workshop on Principles of Engineering Service Oriented

Systems (PESOS 2009). hosted at the 31st Int. Conf. on Software Engineering (ICSE 2009), IEEE Computer Society Press.

[6] A. Arsanjani. Service-oriented modeling and architecture. *IBM developer works*, 2004.

[7] L. Baresi, C. Ghezzi, and S. Guinea. Smart monitors for composed services. In *Proceedings of the 2nd international conference on Service oriented computing*, pages 193–202. ACM New York, NY, USA, 2004.

[8] L. Baresi, C. Ghezzi, and S. Guinea. Towards Self-healing Composition of Services. *Contributions to Ubiquitous Computing: With 131 Figures and 12 Tables*, page 27, 2007.

[9] L. Baresi and S. Guinea. Towards Dynamic Monitoring of WS-BPEL Processes. *LECTURE NOTES IN COMPUTER SCIENCE*, 3826:269, 2005.

[10] G. Booch. Object oriented analysis and design with applications. 1994.

[11] M. Bruno, G. Canfora, M. Di Penta, G. Esposito, and V. Mazza. Using Test Cases as Contract to Ensure Service Compliance Across Releases. *LECTURE NOTES IN COMPUTER SCIENCE*, 3826:87, 2005.

[12] G. Canfora, M. Di Penta, R. Esposito, and M. Villani. A lightweight approach for QoS-aware service composition. In *Proc. 2nd International Conference on Service Oriented Computing (ICSOC04)-short papers*.

[13] G. Canfora, M. Di Penta, R. Esposito, and M. Villani. An approach for QoS-aware service composition based on genetic algorithms. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1069–1075. ACM New York, NY, USA, 2005.

[14] G. Canfora and M. D. Penta. Testing services and service-centric systems: Challenges and opportunities. *IT Professional*, 8(2):10–17, 2006.

[15] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani. Qos-aware replanning of composite web services. *Web Services, IEEE International Conference on*, 0:121–129, 2005.

[16] M. Colombo, E. Di Nitto, M. Di Penta, D. Distante, and M. Zuccala. Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems. *LECTURE NOTES IN COMPUTER SCIENCE*, 3826:48, 2005.

[17] G. Dobson. Owl and owl-s for dependability-explicit service-centric computing. *Service-Oriented Computing: Consequences for Engineering Requirements*, 0:4, 2006.

[18] G. Dobson and I. Sommerville. Disambiguating availability specification through the use of owl. *Service-Oriented Computing: Consequences for Engineering Requirements*, 0:5, 2006.

[19] G. Engels and M. Assmann. Service-Oriented Enterprise Architectures: Evolution of Concepts and Methods. In *Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE*, 2008.

[20] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J. Richter, M. Voß, and J. Willkomm. A Method for engineering a true service-oriented architecture. In *To appear: Proceedings of the 10th International Conference on Enterprise Information Systems. Barcelona, Spain*, 2008.

[21] A. Erradi, S. Anand, and N. Kulkarni. Soaf: An architectural framework for service definition and realization. *Services Computing, IEEE International Conference on*, 0:151–158, 2006.

[22] J. Hutchinson, P. Sawyer, and J. Walkerdine. Is a dominant service-centric sector good for diversity of provision? *Service-Oriented Computing: Consequences for Engineering Requirements*, 0:7, 2006.

[23] N. Kulkarni and V. Dwivedi. The role of service granularity in a successful soa realization ? a case study. *Services, IEEE Congress on*, 0:423–430, 2008.

[24] K. Mahbub and G. Spanoudakis. A framework for requirents monitoring of service based systems. In *Proceedings of the 2nd international conference on Service oriented computing*, pages 84–93. ACM New York, NY, USA, 2004.

[25] K. Mahbub and G. Spanoudakis. Run-time monitoring of requirements for systems composed of web-services: Initial implementation and evaluation experience. *Web Services, IEEE International Conference on*, 0:257–265, 2005.

[26] B. Meyer and C. Mingins. Component-Based Development: From Buzz to Spark. 1999.

[27] M. Papazoglou and W. Van Den Heuvel. Service-oriented design and development methodology. *International Journal of Web Engineering and Technology*, 2(4):412–442, 2006.

[28] M. Papazoglou and W. van den Heuvel. Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal The International Journal on Very Large Data Bases*, 16(3):389–415, 2007.

[29] G. Spanoudakis and K. Mahbub. NON-INTRUSIVE MONITORING OF SERVICE-BASED SYSTEMS. *International Journal of Cooperative Information Systems*, 15(3):325–358, 2006.

[30] G. Spanoudakis, A. Zisman, and A. Kozlenkov. A service discovery framework for service centric systems. *Services Computing, IEEE International Conference on*, 1:251–259, 2005.

[31] A. Tsalgatidou, G. Athanasopoulos, M. Pantazoglou, C. Pautasso, T. Heinis, R. Grønmo, H. Hoff, A. Berre, M. Glittum, and S. Topouzidou. Developing scientific workflows from heterogeneous services. *ACM Sigmod Record*, 35(2):22–28, 2006.

[32] L.-J. Zhang, N. Zhou, Y.-M. Chee, A. Jalaldeen, K. Ponnalagu, R. R. Sindhgatta, A. Arsanjani, and F. Bernardini. Soma-me: a platform for the model-driven design of soa solutions. *IBM Syst. J.*, 47(3):397–413, 2008.