SOPLE-DE: An Approach to Design Service-Oriented Product Line Architectures

Flávio M. Medeiros[1], Eduardo S. de Almeida[2], and Silvio R.L. Meira[1]

[1] Federal University of Pernambuco (UFPE)
[2] Federal University of Bahia (UFBA)
{fmm2,srlm}@cin.ufpe.br, esa@dcc.ufba.br

Abstract. Software reuse is crucial for enterprises interested in software quality and productivity gains. In this context, Software Product Line (SPL) and Service-Oriented Architecture (SOA) are two reuse strategies that share common goals and can be used together to increase reuse and produce service-oriented systems faster, cheaper and customizable to specific customers. In this sense, this work investigates the problem of designing software product lines using service-oriented architectures, and presents a systematic approach to design software product lines based on services. The proposed approach provides guidance to identify, design and document architectural components, services, service compositions and their associated flows. In addition, an initial experimental study performed with the intention of validating and refining the approach is also depicted demonstrating that the proposed solution can be viable.

Keywords: Service-Oriented Architecture (SOA), Software Product Line (SPL), Software Architecture and Software Development Processes.

1 Introduction

Software reuse is a key factor for enterprises interested in reducing development costs and increasing software quality [1]. In this context, SPL and SOA are two reuse strategies that share common goals, i.e., they both support the reuse of existing software and capabilities during the development of new systems and encourage the development of flexible and cost-effective software systems [2].

In this way, SPL and SOA concepts can be used together with the purpose of increasing and systematizing reuse during the Service-Oriented Development (SOD) and producing service-oriented systems faster, cheaper and customizable to specific customers [3]. Moreover, some service characteristics, e.g., dynamic discoverability and binding, can be used to support the development of Dynamic Software Product Lines (DSPL) [4].

This work investigates the problem of designing Service-Oriented Product Line Architectures (SO-PLA). This combination raises several challenges, such as how to identify and design services for the domain, decide the variation points to be considered in the context of SOD, identify service variability implementation mechanisms and define architectural views to represent the SO-PLA.

J. Bosch and J. Lee (Eds.): SPLC 2010, LNCS 6287, pp. 456–460, 2010.

In this sense, a systematic design approach with a set of activities, with clearly defined inputs and outputs, and performed by a predefined set of roles is described in this work with the purpose of providing guidance to solve the problems of designing a SO-PLA. A service-oriented product line is considered as a set of similar service-oriented systems that supports the business processes of a specific domain and can be developed from a common set of core assets [5].

In order to define a SO-PLA, an approach is essential to provide guidance to the team, specify the artifacts to be produced, and associate activities with specific roles and the team as a whole. Without it, the development team may develop software in an ad-hoc manner, with success relying on the efforts of a few dedicated individual participants [6].

2 Related Work

An approach for developing service-oriented product lines was presented in [4]. In this work, a method to identify services and service compositions from feature models is depicted. In our work, we also provide methods to identify service candidates, not only from feature models, but also using business processes, use cases and quality attributes scenarios. In addition, some architectural views are proposed to represent the interactions among architectural elements.

The concept of Business Process Line (BPL) is used in [3]. This work provides a process to develop service-oriented product lines based on business processes that contain variability and can be customized to specific customers. Our work also considers variability in the business processes, but we also use feature models to represent variability in an easy and exploitable way.

In [7], an initial process for service-oriented product lines is presented. This work discusses the characteristics of SOA and SPL processes, but a systematic process for service-oriented product lines is not provided. The key difference of our work is the systematization of design. In addition, our approach was validated and refined through an initial experimental study.

3 The Proposed Approach (SOPLE-DE)

The SOPLE-DE is a top-down approach for the systematic identification, design and documentation of service-oriented core assets supporting the non-systematic reuse of SOD. It is divided in two cycles as SPL engineering. The core asset development cycle aims to provide guidelines and steps to identify, design and document architectural elements with variability. In the product development cycle, the architectural elements are specialized to a particular context according to specific customer requirements [5].

The SOPLE-DE considers the architectural style shown in Figure 1. This architectural style presents the layers that are commonly used in SOA [8]. Thus, SOPLE-DE provides guidelines to identify, design and document architectural elements for these layers. We use this architectural style because we believe that these layers are essential for any SOA solution.

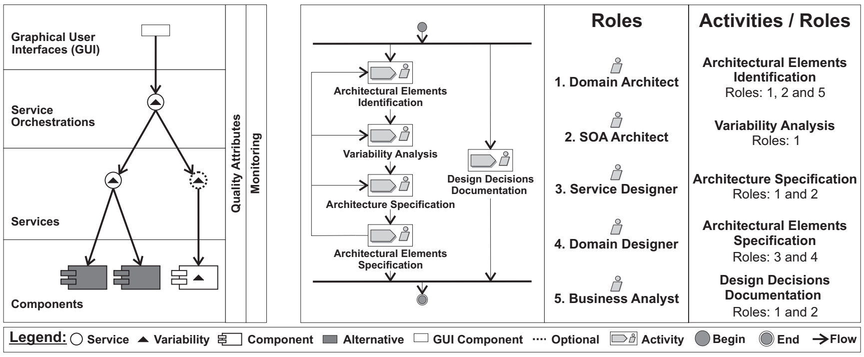

Fig. 1. Layered Architectural Style / SOPLE-DE Activities and Roles

The SOPLE-DE considers that a SO-PLA supports two variability levels as described next [3]: *Configuration variability,* in which architectural elements are selected from the core assets in order to obtain the target system, i.e., optional and alternative architectural elements are selected or excluded; *Customization variability,* in which architectural elements already selected for the architecture are customized according to specific requirements, i.e., architectural elements with variability are customized internally. SOPLE-DE also considers variability in the communication among the architectural elements, e.g., different protocols can be used for communication and the messages exchanged can be sent in a synchronous or asynchronous way.

The activities and roles of the SOPLE-DE are presented in Figure 1. It starts with the architectural elements identification activity, which receives the domain feature model, the business process models and the quality attribute scenarios as mandatory inputs. The domain use cases are optional inputs. It produces a list of components, services and service orchestration candidates for the SO-PLA. Moreover, the communication flows among these elements are also defined.

Subsequently, there is the variability analysis activity. It receives the list of components, services, service orchestrations and their flows identified previously, and defines and documents key architectural decisions regarding variability. In this activity, it is defined how the variability will be implemented. It refines the architectural elements identified previously.

Architecture specification is the next activity, in which the architecture is documented using different views in order to represent the concerns of the different stakeholders involved in the project [9]. An architecture is a complex entity that should be represented and documented upon several views (see Figure 2).

In the architectural elements specification activity, the low-level design of components and services is performed. SOPLE-DE suggests some UML diagrams to document the internal behavior of the architectural elements [10]. In parallel with these four activities described, the design decisions documentation activity is performed concurrently. In this activity, important design decisions, such as the selection of technologies and variability mechanisms, are documented.

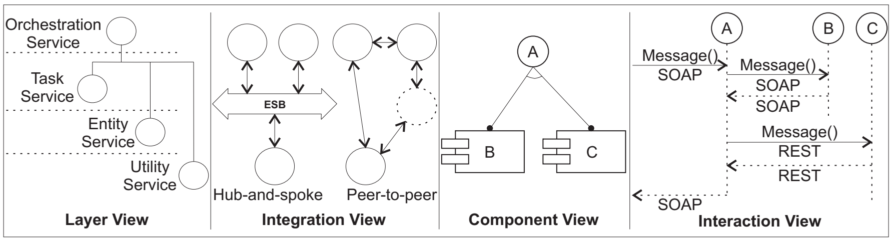

Fig. 2. Architectural Views

4 Experimental Study

An experimental study on the *Travel Reservation* domain was performed with the purpose of evaluating and refining the SOPLE-DE. In this experiment, the process of Wohlin [11] was used to define, plan and execute the experimental study. In addition, the Goal Question Metric (GQM) framework was also used to define the experiment [12]. The goal of this experiment was to *analyze the SOPLE-DE* for the purpose of *evaluation* with respect to its *efficacy* from the point of view of *researcher* in the context of *service-oriented product line projects.*

After collecting the information about the service coupling, instability and cohesion, the data collected was analyzed. Figure 3 shows the metric results for the services identified by the subjects. In the graphics, the axis (X) shows the ID of the subjects, while the axis (Y) represents the service coupling mean, service instability mean and the average cohesion of the service operations.

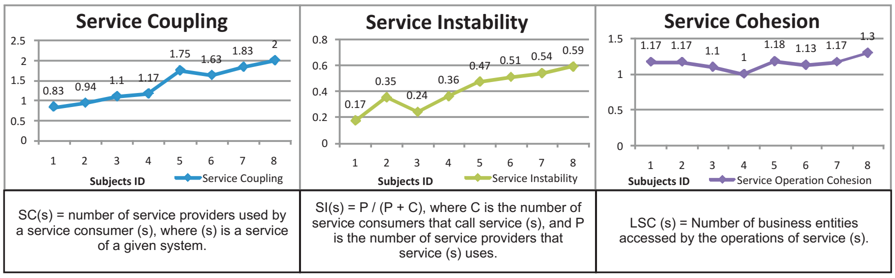

Fig. 3. Metric Results

The subjects with Id = 1, 2, 3 and 4 used the SOPLE-DE during the experiment, while subjects with Id = 5, 6, 7 and 8 designed the project without following a structured method. As it can be seen in Figure 3, the coupling, instability and cohesion of the services generated using the SOPLE-DE are lower when compared with the services identified by the subjects without use the method. The full description of this experiment can be found in [10].

5 Conclusions and Future Work

This work proposed an approach to design service-oriented product lines focusing on increasing reuse and flexibility, and supporting the development of service-oriented systems faster, cheaper and customizable to specific customers.

The SOPLE-DE approach was based on an extensive review of the available service-oriented processes, their weak and strong points and gaps in the area [10]. It can be seen as a systematic way to design service-oriented product line architectures through a well-defined sequence of activities with clearly defined inputs and outputs. Additionally, the approach was evaluated in an experimental study that presented findings that the SOPLE-DE can be viable to aid software architects to design service-oriented product line architectures with good coupling and instability, and identify services with cohesive operations.

Even it being a relevant contribution for the field, new routes need to be investigated in order to define a more complete process that consider all the software development disciplines, such as requirements, design and implementation, for product lines based on services. In addition, new experiments in different domains are necessary to gather more evidences about the efficacy of the proposed approach. Experiments in industry are also considered as future work.

References

1. Krueger, C.W.: Software reuse. ACM Computing Surveys 24(2) (1992)
2. Medeiros, F.M., de Almeida, E.S., Meira, S.R.L.: Towards an approach for service-oriented product line architectures. In: SOAPL 2009 (2009)
3. Boffoli, N., Caivano, D., Castelluccia, D., Maggi, F.M., Visaggio, G.: Business process lines to develop service-oriented architectures through the software product lines paradigm. In: SOAPL, pp. 143–147 (2008)
4. Lee, J., Muthig, D., Naab, M.: An approach for developing service-oriented product lines. In: SPLC, pp. 275–284. IEEE Computer Society, Los Alamitos (2008)
5. Clements, P., Northrop, L.: Software Product Lines: Practices and Patterns. Addison-Wesley, Reading (2001)
6. Booch, G.: Managing the Object-Oriented Project. Addison-Wesley, Reading (1995)
7. Günther, S., Berger, T.: Service-oriented product lines: Towards a development process and feature management model for web services. In: SOAPL (2008)
8. Arsanjani, A.: Service-oriented modeling and architecture. Technical report, Service-Oriented Architecture and Web services Center of Excellence, IBM (2004)
9. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practices. Addison-Wesley Longman Publishing Co., Inc., Boston (2003)
10. Medeiros, F.M.: An approach to design service-oriented product line architectures. Master's thesis, Federal University of Pernambuco (2010)
11. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering: An Introduction. Springer, Heidelberg (2000)
12. Basili, V., Caldiera, G., Rombach, D.H.: The goal question metric approach. In: Encyclopedia of Software Engineering. Wiley, Chichester (1994)