A Systematic Approach to Service-Oriented Analysis and Design*

Soo Ho Chang and Soo Dong Kim

Department of Computer Science
Soongsil University, Seoul, Korea
shchang@otlab.ssu.ac.kr, sdkim@ssu.ac.kr

Abstract. Service-oriented computing (SOC) has several unique features which are not typically presented in conventional object-oriented development (OOD) and component-based development (CBD), including the commonalty of service functionality, publish and discovery paradigm of services and dynamic composition of possibly 3rd party's independent unit services. Hence, OOD and CBD-based analysis and design methods are not effective and expressive enough to model service-oriented applications. Rather, Service-Oriented Analysis and Design (SOAD) has to be defined by using some of the two modeling paradigms and by adding SOC-unique modeling mechanisms. In this paper, we first present a technical comparison among OOD, CBD and SOAD to derive the design criteria for SOAD. And, we define the key artifacts that have to be delivered during SOAD. Based on this, we present a SOAD process which takes service requirements as the initial input and delivers service specifications, compositions, and verified service components as the final deliverables. Each of the five phases in the process is defined with its overview, artifacts, and work instructions. Finally, we present a case study of applying our process to a service domain to reveal its effectiveness and applicability. Once the proposed SOAD is well applied, SOAD artifacts can be more systematically and efficiently implemented with current SOA standards.

1 Introduction

Service-Oriented Computing (SOC) is emerging as a new paradigm for building and maintaining applications in a cost effectiveness way. With increased support with SOA tools, the paradigm of *publish-discover-compose* services becomes more common, and it has significant benefits for ROI perspective as well as technical perspective. However, there is a considerable amount of gap between the promises of SOC and the maturity of service engineering methodology. The main cause of the gap is the lack of effective analysis and design methods. That is, there is a great demand on effective service-oriented analysis and design (SOAD) methodology [1].

Service-oriented computing (SOC) has several unique features which are not typically presented in conventional object-oriented development (OOD) and component-based

* This work was supported by Seoul Research and Business Development Program (10557).

J. Münch and P. Abrahamsson (Eds.): PROFES 2007, LNCS 4589, pp. 374–388, 2007.

development (CBD). In SOC, services should be designed not just for a dedicated client, but for a family of potential clients. Hence, the commonality of services should be well modeled into service components. Services are not tightly coupled to certain client applications; rather they are published in service repositories, discovered by clients dynamically, and composed to fulfill the functionality expected by clients. Hence, SOAD must provide facilities for supporting SOC-specific features in addition to conventional software modeling facilities.

Since current SOA standards are largely based on the object-oriented paradigm and CBD, existing OOD and CBD methods may seem to be useful for SOAD. However, OOD and CBD methods by themselves are not effective and expressive enough to model service-oriented applications. Rather, SOAD has to be defined with SOC-specific modeling facilities on top of the two existing paradigms.

Hence, our research goal is to define a systematic and effective SOAD which extends OOD and CBD. In this paper, we first present a technical comparison among OOD, CBD and SOAD to better understand their similarities and differences in Section 3. From the comparison, we derive the design criteria for our SOAD method. And, we define the key artifacts that have to be delivered during SOAD in Section 4. Based on these preliminary works, we propose a SOAD process which takes service requirements as the initial input and delivers service specifications, compositions, and verified service components as the final deliverables in Section 5. Each of the five activities in the process is defined with its overview, artifacts, and work instructions. We present a case study of applying our process to a service domain to reveal its effectiveness and applicability in Section 6. Once the proposed SOAD is well applied, SOAD artifacts can be more systematically and efficiently implemented with current SOA standards of WSDL [2], UDDI [3], BPEL[4] and SOAP[5].

2 Related Works

Zimmermann's Work introduces an integration of Business Process Modeling (BPM), Enterprise Architecture (EA), and Object-oriented Analysis and Design (OOAD) [6]. It introduces three major level of abstractions in SOA; operations as single logical units of a work, services as logical groupings of the operations, and business processes. It also represents how the three methods can be applied to service-oriented analysis and design, what should be more defined.

Arsanjani's work presents seven layers of SOA; *operational system layer, enterprise component layer, services layers, business process choreography layer, presentation layer, integration layer, and QoS layer* [7]. For each of these layers, design and architectural decision activities are needed. Especially, they proposes five activities called service identification, service classification or categorization, subsystem analysis, component specification, service allocation, and service realization, which are performed in service-oriented modeling.

Erl addresses a service oriented-analysis and design process with subordinate instructions [8]. The work proposes three steps for service-oriented analysis and five steps for service-oriented design process. Especially, in the service modeling step in service oriented analysis, service candidates, candidate service compositions, application services, and service operations are modeled. Based on the analyzed