International Journal of Web Services Research
Volume 16 • Issue 1 • January-March 2019

Adaptive BPEL Service Compositions via Variability Management:
A Methodology and Supporting Platform

Chang-ai Sun, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China & Science and Technology on Aerospace Intelligent Control Laboratory, Beijing, China

Zhen Wang, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

Ke Wang, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

Tieheng Xue, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

Marco Aiello, University of Stuttgart, Stuttgart, Germany

ABSTRACT

Service-Oriented Architectures are a popular development paradigm to enable distributed applications constructed from independent web services. When coordinated, web services are an infrastructure to fulfill dynamic and vertical integration of business. They may face frequent changes of both requirements and execution environments. Static and predefined service compositions using business process execution language (BPEL) are not able to cater for such rapid and unpredictable context shifts. The authors propose a variability management-based adaptive and configurable service composition approach that treats changes as first-class citizens and consists of identifying, expressing, realizing, and managing changes of service compositions. The proposed approach is realized with a language called VxBPEL to support variability in service compositions and a platform for design, execution, analysis, and maintenance of VxBPEL-based service compositions. Four case studies validate the feasibility of the proposed approach while exhibiting good performance of the supporting platform.

KEYWORDS

Adaptive Systems, Business Process Execution Language, Service Composition, Service Oriented Architectures, Variability Management

INTRODUCTION

Service Oriented Architectures (SOA) are a mainstream development paradigm of distributed applications for open and large networks, like the Internet (Papazoglou et al., 2008; Lazovik et al., 2006). SOA caters for the development of highly-scalable systems, whose elementary units are Web services. Individual Web services often provide limited functionalities; therefore, they are unable to meet in isolation complex requirements. *Service composition* is a powerful mechanism, which

DOI: 10.4018/IJWSR.2019010103

coordinates multiple Web services to support complex business goals. Service compositions manifest a number of interesting features, such as loosely-coupling, standard data exchanging, and protocols-based application invoking. As a consequence, service compositions provide a solid solution for application integration in the distributed, dynamic, and heterogeneous environments, and they are increasingly adopted for system re-engineering and optimization (Sun et al., 2011; Karunamurthy et al., 2012; Cheng et al., 2017).

When service compositions act as an infrastructure to realize dynamic and vertical integration of business processes, they may face frequent changes due to quickly-changing requirements or dynamic environments. Service compositions are expected to be adaptive and configurable, and many efforts have been reported to address this issue (Erradi & Maheshwari, 2005; Ezenwoye & Sadjadi, 2006; Colombo et al., 2006; Hammas et al., 2015; Sheng et al., 2009; Ardagna & Pernici, 2007; Laleh et al., 2017; Yau et al., 2009). One approach is to make service compositions adaptable resorting to the proxy (or broker) pattern (Erradi & Maheshwari, 2005; Ezenwoye & Sadjadi, 2006; Colombo et al., 2006; Hammas et al., 2015). A request message is first sent to a proxy, the message is then passed to the targeted Web service, finally the targeted Web service executes the required operations and returns the result to the proxy, and the proxy transfers the result to the invoker. This approach focuses on service selection or replacement in service composition instances at run-time. Another approach is based on annotations (Sheng et al., 2009; Ardagna & Pernici, 2007; Laleh et al., 2017). Constraints, policies or rules of service compositions are specified in a declarative manner, and then predefined goals are satisfied.

A recent trend is to address the adaptation issue via modelling the variability of service compositions (Nguyen et al., 2011; Galster & Avgeriou, 2013; Wang & Wang, 2013; Alférez et al., 2014; Nguyen et al., 2016). There are two ways, namely, imperative and declarative (Aiello et al., 2010). The imperative method focuses on the procedure of constructing a process and defining fixed sets of variations. For instance, Provop is a framework capturing all process variants in a single process model, which defines the optional element to implement the changes of the basic process model (Hallerbach et al., 2008). The declarative method focuses on expressing the logic of a process without explicitly describing its control flows. Pesic et al. (2007) proposed an approach to support variability of workflow systems via constraint-based models. Changes can be done by handling constraints and activities at run-time. Service compositions resemble workflow systems in the context of SOA, while manifesting many new features. Supporting variability of service compositions is still not mature and significant efforts are required in this direction.

In our previous work (Koning et al., 2009; Sun et al., 2010a), we focused on executable service compositions with variability design. We have developed *VxBPEL* (Koning et al., 2009) by extending BPEL (Eviware, 2007). With VxBPEL, one can specify variation points, variants, and realization relations in a service composition (Sun et al., 2010a). To facilitate the adoption of VxBPEL in practice, we developed an engine (Sun et al., 2013) and an analysis tool (Sun et al., 2010b) to execute and analyze service compositions that are specified in VxBPEL, respectively. Although VxBPEL has the ambition to provide adaptable service compositions implemented in BPEL, the following issues remain open. First, a framework or methodology that can be used to systematically identify, design, implement, and maintain variability of service compositions is still missing. Second, the current version of VxBPEL only provides a basic way for specifying variation points and variants without supporting complex dependencies among multiple variation points or variants, which are essential to complex, realistic business processes. Third, a platform that fully supports the variability management-based service composition framework is still not available.

In this paper, we propose a variability management-based approach to systematically address the adaptation and re-configuration issues of service compositions. The approach provides a unified framework with an extended language to identify, specify, realize, and manage changes of service compositions, and the framework is enabled with a comprehensive supporting platform. The novel contributions that go beyond those of our previously published work (Koning et al., 2009; Sun et al., 2010a; Sun et al., 2010b; Sun et al., 2013) are:

Towards Event-Driven Context-Aware Web Services

Guadalupe Ortiz, Juan Boubeta-Puig, Alfonso García de Prado and Inmaculada Medina-Bulo (2013). *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions (pp. 148-159).*

www.igi-global.com/chapter/towards-event-driven-context-aware/69472?camid=4v1a

Broadening JAIN-SLEE with a Service Description Language and Asynchronous Web Services

Paolo Falcarin, Claudio Venezia and José Felipe Mejia Bernal (2010). *Web Services Research for Emerging Applications: Discoveries and Trends (pp. 442-464).*

www.igi-global.com/chapter/broadening-jain-slee-service-description/41533?camid=4v1a