

Chapter 73

Towards a High- Availability-Driven Service Composition Framework

Jonathan Lee

National Central University, Taiwan

Shin-Jie Lee

National Central University, Taiwan

Shang-Pin Ma

National Taiwan Ocean University, Taiwan

Chia-Ling Wu

National Central University, Taiwan

Chiung-Hon Leon Lee

Nanhua University, Taiwan

ABSTRACT

Service-Oriented Computing (SOC), a main trend in software engineering, promotes the construction of applications based on the notion of services. SOC has recently attracted a great deal of attention from researchers, and has been comprehensively adopted by industry. However, service composition enabling the aggregation of existing services into composite services still imposes a great challenge to service-oriented technology. Web service composition requires component Web services to be available in request, to behave correctly in operation, and to be replaceable flexibly in failure. Although availability of Web services plays a crucial role in building robust SOC-based applications, it has been largely neglected, especially for service composition. In this chapter, we propose a service composition framework that integrates a set of composition-based service discovery mechanisms, a user-oriented service delivery approach, as well as a service management mechanism for composite services.

INTRODUCTION

Service-Oriented Computing (SOC) has become a main trend in software engineering that exploits both Web services and Service-Oriented Architecture (SOA) as fundamental elements for developing

on-demand applications. Web services are self-described, self-contained, platform-independent computational elements that can be published, discovered, and composed using standard protocols to build applications across various platforms and organizations. Recently, research in both industry and academia has alleviated issues in service-oriented technologies, including semantic service

DOI: 10.4018/978-1-4666-4301-7.ch073

discovery, automatic service composition, and QoS aware service management. Notably the composition of multiple Web services into a composite service has been widely accepted as more beneficial to users than an atomic service. In further service compositions to service clients, resulting composite services can be used as atomic services. However, Web service composition requires component Web services to be available and to behave correctly. In fact, if any component Web service fails, the entire service flow must be aborted. This shows the importance of including mechanisms of maintaining service availability in Web service compositions. Although Web service availability is a critical issue for building robust SOA-based applications, it has been largely neglected, especially for service composition. Current specifications like WS-ReliableMessaging and WS-Transaction do not focus on the availability of Web services (Abraham, 2005). WS-Reliability guarantees that a message is delivered to a recipient even if the recipient is down temporarily. Similarly WS-Transaction guarantees that a client does not stay in an invalid state when any Web service in the transaction is not operational. Accordingly, we still need a systematical methodology to enhance or maintain service availability, especially for composite services. To achieve the goal, we have identified three key issues:

- **How to handle the openness issue of Web services?** The implementation of the service-oriented architecture can be classified as two categories, one for the intranet and the other for the Internet. For the SOA implementations within the intranet, the organization-level management administers all component services, therefore, existing methods of system availability assurance, such as load balance, failover mechanism, or even manual control by the system management personnel, can attain the required service availability. However, the SOA implementations over the Internet imply that all services reside in an open environment and component services in a composition may belong to multiple different service providers from different organizations. Thus, maintaining the service availability through central control and management is arduous, increasing the difficulty of achieving a high service availability of composite services.
- **How to handle the dynamic issue (Cervantes, 2005) of Web services?** As mentioned above, most Web services reside in an open environment. The large number of services together with their communication links and information sources may appear and disappear dynamically in the Internet. Moreover, Web services may suffer from long response times or temporary non-availability. Such outages for mission critical systems are unacceptable and can even cause fatal damages. Handling the dynamic issue, that is, detecting the permanently or temporarily failed services and finding required services that can be re-bound and react appropriately in a composition, is significant to providing dependable SOA-based applications.
- **How to decide service recovery strategies?** When a composite service detects any failure from its component service, several possible remedies may be applied to, including some intuitive solutions such as restarting failed service directly and re-invoking component services after waiting for a period of time, or a far more complicated design of substituting a functionally equivalent service for the failed one. Swapping services relies on a combination of various search strategies, including searching for services of better quality, services that many users use, or client-specified services. A mechanism is in need to assist the service manager in choosing the most appropriate recovery strategy to satisfy managers' expectations or users' demands.

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/towards-high-availability-driven-service/77768

Related Content

The Method and Tools Development for Web-Cameras Color Correction in Binocular Vision Systems

Konstantin Dergachov, Leonid Krasnov, Olexander Cheliadinand Olexander Plakhotnyi (2021). *Handbook of Research on Software Quality Innovation in Interactive Systems* (pp. 154-191).

www.irma-international.org/chapter/the-method-and-tools-development-for-web-cameras-color-correction-in-binocular-vision-systems/273569

Failure Prediction Approach in Agile Software Development

Bulqees Alajaleenand Aysh Alhroob (2022). *International Journal of Software Innovation* (pp. 1-11).

www.irma-international.org/article/failure-prediction-approach-in-agile-software-development/292025

DSLs in Action with Model Based Approaches to Information System Development

Ivan Lukovic, Vladimir Ivancevic, Milan Celikovicand Slavica Aleksic (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 596-626).

www.irma-international.org/chapter/dsls-action-model-based-approaches/77724

Hardware-In-the-Loop Testing of On-Board Subsystems: Some Case Studies and Applications

Luca Pugiand Benedetto Allotta (2012). *Railway Safety, Reliability, and Security: Technologies and Systems Engineering* (pp. 249-280).

www.irma-international.org/chapter/hardware-loop-testing-board-subsystems/66675

Ontology Alignment Quality: A Framework and Tool for Validation

Jennifer Sampson, John Krogstieand Csaba Veres (2011). *International Journal of Information System Modeling and Design* (pp. 1-23).

www.irma-international.org/article/ontology-alignment-quality/55486