Chapter 12

Goal–Based Requirements Elicitation for Service Reuse in Cloud Computing

Lin Liu
Tsinghua University, China

ABSTRACT

Based on the existing requirements model and new service requests, the proposed service requirements elicitation framework ASREGL aims to achieve an optimal service supply and demand relationship. The current control variable is the similarity of the service requirements and capabilities. The major contents of this chapter include: (1) Service requirements elicitation questionnaires system, which uses a set of queries to narrow generic service requirements down to specific expressions of user preferences; (2) The construction process of goal-oriented service models based on the user’s answers to the elicitation questions, which incorporates a Multi-strategy Task-adaptive Learning (MTL) technique with the goal-oriented requirements modelling approaches such as the i* framework; and (3) The dual feedback control mechanism, which supports the continuous interactions between service providers and requestors. The requirements elicitation process uses requirements models in the services pool matched with specific service requests, and ultimately identifies a service capability model that best fits the specific purpose of the targeted user. A virtual travel agency example is used to illustrate the proposed approach.

DOI: 10.4018/978-1-4666-0897-9.ch012
1. INTRODUCTION

The Service-Oriented Computing paradigm aims to support maximal reuse of web services according to the user’s requirements. This target becomes even more viable in the cloud computing platform. At present, the user’s requirements are often represented in certain existing standard interoperable service description languages such as WSDL or OWL-S. However, general service requestors may find such languages hard to use directly because service requirements are often partially understood and elicited. This chapter formulates the service-oriented requirements analysis as an automated feedback control process, in which the classical “once for all” philosophy is replaced with a continuous learning, negotiation and adaptation process. Conventional requirements analysis is a collaborative negotiate-and-decide process carried out by requirements analysts and system-to-be stakeholders. The requirements analysts usually drive requirements-related activities including elicitation, analysis and prioritization, addressing questions such as: What information shall be collected from each stakeholder? In what order should questions be asked? How to settle on a comprehensive list of functional requirements? How to settle on a comprehensive list of desired qualities of service? How to minimize the number of questions asked, while ensuring that adequate information is elicited from stakeholders? How to adjust the elicitation process according to answers obtained from earlier questions? The research discipline of Requirements Engineering (RE) has accumulated a body of knowledge that can guide an analyst in going through the requirements elicitation process. Of course, the analyst’s intuitions and experiences also play a vital role.

During recent years, the Service-Oriented Architecture (SOA) paradigm (Papazoglou et al. 2007) has been recognized as a novel way to structure complex, loosely-coupled services on demand and timely. Among other advantages, SOA can adapt to changes rapidly, and provide effective inter- and intra-organization application integration. It is worth mentioning that, no matter how neatly the SOA defined the relationships between the services and the responsibility of each one, the service itself has to be built and runs somewhere. Cloud Computing embraces various techniques of networking, virtualization, software as a service (SaaS), Web 2.0, web infrastructure, distributed computing, and also Services Computing. Cloud Computing platforms provide large scalable computing resources “as a service” over the Internet to users (Peng et al. 2009). Cloud Computing has drawn more and more attention from research as well as industry community. There are some basic techniques of Cloud Computing, for example Bigtable (Chandra et al. 2006), Google File System (GFS) (Ghemawat et al, 2003), and MapReduce (Dean et al. 2008). There are some open-source Clouding Computing infrastructure such as Xen Cloud Platform (XCP), Nimbus, OpenNebula (Sotomayor et al. 2009), and Eucalyptus. Any earlier enterprise software architectures prior to cloud computing will lead to capacity problems, or chaotic situations. The estimation of capacity for business applications is challenging, estimating the capacity requirements for SOA services, especially if it is a large quantity of service, is even more challenging. Thus, cloud computing and SOA come together. Services defined within an SOA are best deployed within a cloud to gain all of the advantages of cloud-based software (Marks & Lozano, 2010). On the other hand, cloud applications are at their best when defined in terms of services. The two are inseparable.

SOA are founded on the concept of service. A service is something one actor wants and another can deliver, along with terms constraining the delivery. Service-level agreements (SLAs) and quality-of-service (QoS) requirements are two generic types of service terms. An operational framework for services actually involves three kinds of actors: requestors searching for needed services, providers who provide services and
Related Content

An Integrated Approach for Specification and Analysis of Functional and Performance Properties of Concurrent Systems

Situational Fit in Incremental Method Engineering
[www.igi-global.com/article/situational-fit-incremental-method-engineering/70924?camid=4v1a](www.igi-global.com/article/situational-fit-incremental-method-engineering/70924?camid=4v1a)

Web Application Vulnerabilities and Their Countermeasures
[www.igi-global.com/chapter/web-application-vulnerabilities-and-their-countermeasures/188258?camid=4v1a](www.igi-global.com/chapter/web-application-vulnerabilities-and-their-countermeasures/188258?camid=4v1a)

Goal Modelling for Security Problem Matching and Pattern Enforcement
[www.igi-global.com/article/goal-modelling-for-security-problem-matching-and-pattern-enforcement/201215?camid=4v1a](www.igi-global.com/article/goal-modelling-for-security-problem-matching-and-pattern-enforcement/201215?camid=4v1a)