Multi-Level Modeling of Web Service Compositions with Transactional Properties

K. Vidyasankar, Memorial University, Canada
Gottfried Vossen, University of Muenster, Germany

ABSTRACT

Web services have become popular as a vehicle for the design, integration, composition, reuse, and deployment of distributed and heterogeneous software. However, although industry standards for the description, composition, and orchestration of Web services have been under development, their conceptual underpinnings are not fully understood. Conceptual models for service specification are rare, as are investigations based on them. This paper presents and studies a multi-level service composition model that perceives service specification as going through several levels of abstraction. It starts from transactional operations at the lowest level and abstracts into activities at higher levels that are close to the service provider or end user. The authors treat service composition from a specification and execution point of view, where the former is about composition logic and the latter about transactional guarantees. Consequently, the model allows for the specification of a number of transactional properties, such as atomicity and guaranteed termination, at all levels. Different ways of achieving the composition properties and implications of the model are presented. The authors also discuss how the model subsumes practical proposals like the OASIS Business Transaction Protocol, Sun’s WS-TXM, and execution aspects of the BPEL4WS standard.

Keywords: Multi-Level Transactions, Service Composition, Service Orientation, Transactional Properties, Web Service

INTRODUCTION

Web services (Alonso et al., 2004; Casati & Dayal, 2002) have become popular as a vehicle for the design, integration, composition, reuse, and deployment of distributed and heterogeneous software, based on the premise that distributed computing can be made a reality easier than with previous approaches such as RPC, object-orientation, or static middleware. However, while industry standards for the description, composition, and orchestration of Web services have been under development for quite some time already, their conceptual underpinnings are still not fully understood. Indeed, conceptual models for service specification are rare, as are investigations based on them. This paper tries to make a contribution in this direction. In particular, it presents a multi-level service composition model that perceives
service specification as a process that goes through several levels of abstraction: It starts from transactional concepts at the lowest level, and then gradually abstracts into activities at higher levels that are close to the service provider or even the end user. Importantly, the model allows for a specification of desirable composition properties such as atomicity and guaranteed termination at all levels, thereby encompassing a number of real-world approaches such as the OASIS BTP or Sun’s WS-TXM.

Web services are currently seen by software vendors and application developers as an appropriate way of handling both application and data integration problems. The general vision is that services can be described in an implementation-independent fashion, and that users or clients can compose such descriptions into new services and execute the results by referring back to the respective service providers. To achieve these goals, a variety of industry standards have become available, most notably SOAP (Simple Object Access Protocol) for transportation purposes (Newcomer, 2002), WSDL (Web Services Description Language) for service descriptions (Newcomer, 2002), and BPEL4WS Business Process Execution Language for Web Services (IBM, 2007) for the description of service compositions in the form of graph-based process models.

Second, Web services represent an important way of realizing a so-called service-oriented architecture (SOA) (Huhns & Singh, 2005; Singh & Huhns, 2005). An SOA tries to answer the question of which services are available within a given enterprise or application context, which ones need to be newly implemented, and which ones need to be obtained from an external provider. To this end, it is reasonable to assume that, from a top-down development perspective, it makes sense to come up with one or more process models that clarify and fix the goals and procedures clients want to support by appropriately chosen services. Such models will typically be tied to a particular application domain, such as commerce, banking, the travel industry, etc. and their components will ultimately be mapped to services. The result will then be an architecture fixing the composition and integration details at a conceptual level (Vossen, 2006).

As has been noted, for example, by Hull et al. (2003), the conceptual underpinnings of Web services are still not fully understood. For example, in BPEL4WS it is possible to define choreographies (or service compositions) by defining a flow of control using guarded links between the respective activities (which appear in <flow> tags); yet this is entirely syntactic, and there is no way to argue about the properties of the resulting flow. On the other hand, studies such as Hull et al. (2003) indicate that service composition may be more intricate than what the standardization committees assume. Using models such as Mealy automata, Hull and others have been able to show that undesirable side effects may occur when certain types of services are composed (e.g., the result of composing “regular” services may all of a sudden be a “context-sensitive” service).

The model we are proposing and studying in this paper is based on the perception that service composition is not adequately described as long as flat models are used; instead, in a flat model, be it classical transactions, finite-state automata, or Petri nets, the composition designer has to fix a particular level of abstraction and then will run into difficulties when trying to argue about properties that relate to (lower-level) components or to (higher-level) aggregations and that hence actually span several logical levels of the composition. Opposed to this, our intention is to construct a “bridge” between a low-level model that is based on classical transactions (Weikum & Vossen, 2002), a model that generalizes transactional guarantees to an (intermediate) process level (Schuldt et al., 2002), and a high-level model such as the ones used in PARIDE (Mecella et al., 2002) that orchestrates e-services via Petri nets.

A Service Composition Example

As a motivating example, we consider an electronic shopping scenario, where a customer is hunting for some specific goods (such as a
www.igi-global.com/article/understanding-business-domain-models/49724?camid=4v1a

A Tool for Fuzzy Reasoning and Querying
www.igi-global.com/chapter/tool-fuzzy-reasoning-querying/20361?camid=4v1a