

## Chapter 6

# Transactional–Aware Web Service Composition: A Survey

**Yudith Cardinale**

*Universidad Simón Bolívar, Venezuela*

**Joyce El Haddad**

*Université Paris-Dauphine, France*

**Maude Manouvrier**

*Université Paris-Dauphine, France*

**Marta Rukoz**

*Université Paris-Ouest Nanterre La Défense & Université Paris-Dauphine, France*

### **ABSTRACT**

*Web Service (WS) composition consists in combining several WSs into a Composite WS (CWS), which becomes a value-added process. In order to provide reliable and fault-tolerant CWSs, several transactional-aware composition approaches have been proposed. However, as far as we know, no real classification survey of such approaches exists. This is the contribution of this chapter. Our classification distinguishes the more relevant and recent propositions in two groups: approaches based on WS transactional properties and the ones also integrating QoS criteria to the composition process. All these studied approaches are compared according to several criteria: the transactional model used or proposed, the control flow model used or automatically generated, the mechanism proposed to verify the transactional property of the composition, the step(s) of the composition process involved in, and the protocols or the standard languages used or extended. This classification allows underlining the lacks and the future directions which should be studied.*

DOI: 10.4018/978-1-61350-432-1.ch006

## **INTRODUCTION**

Web Services (WSs) are quickly raising as a standard for publishing data and operations for weakly loose, heterogeneous systems, by handling simple request and response messages from/to users to satisfy their requirements. WSs are the most famous implementation of Service-Oriented Architectures (SOA) (Erl, 2005) allowing the construction and the sharing of independent and autonomous software. Flexible SOA connections between services, as well as, software components that pass simple data among other services or that coordinate simple activities, are seen as services which can be combined with other services to achieve specific goals. Thus, SOA provides a scalable and robust framework to integrate heterogeneous software agents and enhance reliability of isolated software components (Papazoglou et al., 2007).

As WSs proliferate, it becomes more difficult to find a specific service that can perform a given task, and combinations of several services may be required to satisfy the more complex user queries, which claim for functional and non-functional requirements. SOA technology trend proposes that applications should not be manually developed but integrated from an adequate set of already existing and available WSs. Successive compositions of WSs usually create a complex structure of interactions among a large number of Composite WSs (CWSs) distributed over the Internet. In this sense, automatic service composition has a great potential to facilitate the integration of WSs deployed in distributed platforms (Rao & Su, 2004). In this context, Web Service Composition (WS Composition) consists in aggregating pre-existing WSs, developed by different organizations and offering diverse functional (e.g. ticket purchase, information search), Quality of Service (e.g. execution time, price), and transactional (e.g. compensable or not) properties, to produce a CWS. It allows building, what is called by Ben

Lakhal et al. (2009), value-added services or more powerful and feature-rich processes.

Web Service Selection corresponds to the step of the WS Composition process where the most appropriate service is selected for each operation from a set of candidates which are differentiated by their non-functional properties (Li et al., 2009). The selection task can be static or dynamic. As explained by Egambaram et al. (2010), the differentiation between both selection techniques deals with the moment at which a concrete WS is integrated into the CWS. With static selection the concrete services are determined and integrated into the CWS at design time. With dynamic selection on the other hand, at design-time there is only a specification of the type of required service and the concrete WS is then integrated at run-time. The order in which WSs should be executed is called the control flow structure of the CWS. This control structure can be generated automatically in the selection step (Shin et al., 2009) or can be generated “by hand” by the designer/user by specifying an abstract representation of processes (e.g., a workflow, a Petri Net) (Huang et al., 2009).

After the selection step, the CWS should be executed over the distributed environment; however the interoperation of distributed software-systems is always affected by failures, dynamic changes, availability of resources, user errors, or interference of concurrently executing actions. In this context, Liu et al. (2010) consider that delivering reliable CWS over unreliable services is a challenging problem. A service that does not provide a proper transaction management support that prevents the system from entering undesired states might be as useless as a service not providing the desired functional results (Cardinale et al., 2010). If the composition is based on WSs by only considering functional requirements and QoS properties, then it is possible that during the execution, the whole system becomes inconsistent in presence of failures. Thus, considering transactional properties in WS Composition allows the system to guarantee reliable composition execution. Indeed,

24 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/transactional-aware-web-service-composition/60884](http://www.igi-global.com/chapter/transactional-aware-web-service-composition/60884)

## Related Content

---

### Economic Decision-Making and the Impact of Risk Management: How They Relate to Each Other

Brian J. Galliani and Gabrielle Battiloro (2019). *International Journal of Service Science, Management, Engineering, and Technology* (pp. 1-13).

[www.irma-international.org/article/economic-decision-making-and-the-impact-of-risk-management/228153](http://www.irma-international.org/article/economic-decision-making-and-the-impact-of-risk-management/228153)

### Resource Management Mechanisms to Support SLAs in IaaS Clouds

David Breitgand, Amir Epstein and Benny Rochwerger (2012). *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice* (pp. 288-307).

[www.irma-international.org/chapter/resource-management-mechanisms-support-slacs/66238](http://www.irma-international.org/chapter/resource-management-mechanisms-support-slacs/66238)

### Generations of Technology Intelligence in the SMEs: A Science Park Case Study

Mahdi Hajari, Mahdi Hamidi and Alireza Aslani (2015). *International Journal of Service Science, Management, Engineering, and Technology* (pp. 50-62).

[www.irma-international.org/article/generations-of-technology-intelligence-in-the-smes/127976](http://www.irma-international.org/article/generations-of-technology-intelligence-in-the-smes/127976)

### Critical Business Model Issues in Deploying NFC Technology for Mobile Services: Case Mobile Ticketing

Antero Juntunen, Virpi Kristiina Tuunainen and Sakari Luukkainen (2012). *International Journal of E-Services and Mobile Applications* (pp. 23-41).

[www.irma-international.org/article/critical-business-model-issues-deploying-nfc/68828](http://www.irma-international.org/article/critical-business-model-issues-deploying-nfc/68828)

### Privacy Management Architecture Privacy Technologies

Larry Korba, Ronggong Song and George Yee (2006). *Privacy Protection for E-Services* (pp. 234-264).

[www.irma-international.org/chapter/privacy-management-architecture-privacy-technologies/28143](http://www.irma-international.org/chapter/privacy-management-architecture-privacy-technologies/28143)