# Chapter 8
# Ensuring Customised Transactional Reliability of Composite Services

**Sami Bhiri**
*National University of Ireland, Galway, Ireland*

**Walid Gaaloul**
*Telecom & Management SudParis, France*

**Claude Godart**
*LORIA-INRIA, France*

**Olivier Perrin**
*LORIA-INRIA, France*

**Maciej Zaremba**
*National University of Ireland, Galway, Ireland*

**Wassim Derguech**
*National University of Ireland, Galway, Ireland*

## ABSTRACT

*Web services are defined independently of any execution context. Due to their inherent autonomy and heterogeneity, it is difficult to examine the behaviour of composite services, especially in case of failures. This paper is interested in ensuring composite services reliability. Reliable composition is defined as a composition where all instance executions are correct from a transactional and business point of view. In this paper, the authors propose a transactional approach for ensuring reliable Web service compositions. The approach integrates the expressivity power of workflow models and the reliability of Advanced Transactional Models (ATM). This method offers flexibility for designers to specify their requirements in terms of control structure, using workflow patterns, and execution correctness. Contrary to ATM, the authors start from the designers' specifications to define the appropriate transactional mechanisms that ensure correct executions according to their requirements.*

## INTRODUCTION

The Web services approach (Erickson & Siau, 2008) has gained broad acceptance as a key technology to automate Business-to-Business (aka B2B) interactions. The Web services technology is an implementation of Service Oriented Architecture[1]. A Web service can be defined as an independent and auto descriptive software component that can be dynamically discovered and invoked through the Internet (or Intranet).

Companies can encapsulate their business processes and publish them as services; and

DOI: 10.4018/978-1-4666-2044-5.ch008

afterwards discover, share and aggregate them. One interesting feature is the possibility to create added value services by composing other services. It is worth noting that the term composite service is usually used to denote composition of operations offered by different services (Alonso, Casati, Kuno, & Machiraju, 2004; Jordan et al., 2007). Throughout this paper, and without loss of generality, we consider services offering only one operation.

Different from business process components, services are generally provided by different organisations independently of any execution context. Since each organisation has its own rules, Web services are treated like strictly autonomous entities. Due to their inherent heterogeneity and autonomy it is difficult to predict the behaviour of composite services especially in case of failures.

The problem we tackle in this paper is how to ensure composite services reliability. By reliable composition, we mean every composition where all its instances have correct execution. An execution is said to be correct if it performs and finishes as intended (from a business point of view). In other terms, an instance execution is correct if it reaches its objectives and completes successfully or it fails properly in a consistent state according to designers requirements.

Current Web service technologies are unable to solve this problem efficiently. These technologies rely mainly on two strong approaches: Workflow Systems (WfS for short) (Georgakopoulos, Hornick, & Sheth, 1995; Jablonski & Bussler, 1996; Leymann & Roller, 1999; van der Aalst & van Hee, 2002; Fischer, 2007) and Advanced Transactional Models (ATM for short) (Georgakopoulos, Hornick, Krychniak, & Manola, 1994; Elmagarmid, Leu, Litwin, & Rusinkiewicz, 1990; Elmagarmid, 1992; Garcia-Molina & Salem, 1987; Garcia-Molina et al., 1991; Moss, 1981; Weikum & Schek, 1992). Taken separately, these technologies are unable to resolve composite services reliability in an effective way.

Indeed, while Workflow management is the key technology for automating business processes enabling their modelling, enactment and analysis, they don't consider reliability problems in case of task failures. On the other hand, ATM ensures reliability of advanced transactions by extending the flat structure of the ACID transactional model and relaxing atomicity and isolation properties. Nevertheless, ATM remains limited for supporting workflow-like applications. First, their control structure still remain primitive compared to those of business processes and consequently to those of composite services. Secondly, they impose a set of constraints regarding the structure and transactional semantics that designers must comply to when modelling their applications.

In this paper, we propose a model for composite Web services that combines the expressivity power of workflow models and the reliability of transactional models. Based on this model, we present a set of techniques enabling to ensure reliability according to designers' requirements (in terms of control flow and failure atomicity). We use the Accepted Termination States (ATS for short) (Elmagarmid et al., 1990; Rusinkiewicz & Sheth, 1995) property as an execution correctness criterion to relax atomicity.

The originality of our approach is the flexibility offered to designers to specify their requirements in terms of control flow and correction (by specifying the set of ATS). Contrary to ATM, we start from designers' specifications to define transactional mechanisms allowing ensuring correct executions according to their requirements. To the best of our knowledge, and as stated in (Rusinkiewicz & Sheth, 1995), "defining a transaction with a particular set of properties" (ATS in our case) "and assuring", a priori, that every execution "will preserve these properties remains a difficult problem" (p. 3).

The remainder of the paper is organised as follows. We first motivate our work through an example which we use throughout the paper. Then we present our transactional Web service model