A Reservation-Based Extended Transaction Protocol for Coordination of Web Services within Business Activities

Wenbing Zhao
Cleveland State University, USA

Firat Kart
University of California, Santa Barbara, USA

L. E. Moser
University of California, Santa Barbara, USA

P. M. Melliar-Smith
University of California, Santa Barbara, USA

ABSTRACT

This chapter describes a novel reservation-based extended transaction protocol for coordination of tasks within a business activity. With the advance of Web Services, it is anticipated that the implementation of business activities as Web Services, and the automation of business activities across multiple enterprises over the Internet, will become a reality. Classical extended transaction protocols are not well suited for this new breed of business activity, because of their use of compensating transactions. The reservation-based extended transaction protocol, described in this chapter, eliminates the need for compensating transactions by using an explicit reservation phase and an explicit confirmation/cancellation phase. This chapter provides a mapping of the reservation-based extended transaction protocol onto the Web Services Coordination Specification, and describes an implementation of the protocol.

INTRODUCTION

The Service Oriented Architecture and Web Services make it possible for enterprises to conduct business over the Internet in real-time without human supervision or intervention (Booth, Haas et al. 2004, Moser and Melliar-Smith 2009). Such business operations are often organized as business activities. A business activity consists of one or more related tasks that are carried out over a potentially long period of
time in a loosely-coupled distributed environment. Such direct and automated computer-to-computer interactions can provide speed improvements and cost reductions for many enterprises.

Because of the nature of business activities, some extended transaction protocols, most notably sagas (Garcia-Molina and Salem 1987), are often used to coordinate the tasks of a business activity, where each individual task is executed as a classical transaction that satisfies the ACID properties, namely, Atomicity, Consistency, Isolation and Durability (Gray and Reuter 1993). If an error occurs during a business activity, the entire business activity is rolled back by applying one or more compensating transactions to reverse the committed transactions. Although useful in many cases, compensating transactions have their limitations. One problem is the cascading compensations that result from relaxation of the isolation property, i.e., before the compensating transaction is applied, other transactions might see the results of the committed transaction and, therefore, must also be compensated. Identifying such transactions is difficult, because there is no way to find them a priori. Furthermore, it might be difficult or impossible to compensate a committed transaction. For example, if an end-of-quarter audit transaction is executed immediately after a sales task is committed in a publicly traded company, the sales are included in the total revenue reported to the public. The completion of the audit transaction, followed by the compensation of the sales task, can result in inconsistencies. In general, the programming of compensating transactions is very difficult and prone to error.

In this chapter, we present a novel reservation-based extended transaction protocol that avoids the use of compensating transactions while achieving atomicity and consistency similar to or better than other existing extended transaction protocols. Each task within a business activity is executed as two steps. The first step involves an explicit reservation of resources according to the business logic. For example, if the task involves reserving two seats, out of 200 available seats, on an airline flight, those two seats are explicitly reserved in a separate step. In the interests of the airline, a fee that is proportional to the duration of the reservation can be associated with the reservation. The second step involves the confirmation or cancellation of the reservation. Each of the steps is executed as a separate traditional short-running transaction, as in the sagas strategy (Garcia-Molina and Salem 1987). However, because of the explicit reservation, other transactions cannot interfere with the business activity. Thus, degree 3 isolation (Gray and Reuter 1993) is achieved and, if a business activity must be abandoned, recovery involves only the cancellation of the reservations for participants in that business activity.

In subsequent sections, we show how the reservation-based extended transaction protocol (hereafter referred to as the Reservation Protocol) can be implemented on top of the Web Services Coordination Specification, in a similar manner to WS-C and WS-BA (Cabrera, Copeland et al. 2005a, 2005c). We describe our implementation of the Reservation Protocol as middleware libraries that are linked into the application processes at the client and the server. We compare the response time, throughput and completion time of the Reservation Protocol with that of the Two Phase Commit Protocol and the Optimistic Two Phase Commit Protocol, obtained from experimental measurements based on our implementations of the protocols.

**SYSTEM MODEL**

A business activity is a unit of work that spans two or more enterprises and consists of one or more tasks. A task is a short-duration unit of work that is executed as a traditional transaction within a single enterprise. The tasks in a business activity are partially ordered. Tasks that are not causally related can be executed concurrently, and causally related tasks are executed according to the partial order.