Chapter 7.4
Semantic Web–Enabled Protocol Mediation for the Logistics Domain

Oscar Corcho
Universidad Politécnica de Madrid, Spain

Silvestre Losada
Intelligent Software Components, S.A., Spain

Richard Benjamins
Intelligent Software Components, S.A., Spain

ABSTRACT

Among the problems that arise when trying to make different applications interoperate with each other, protocol mediation is one of the most difficult ones and for which less relevant literature can be found. Protocol mediation is concerned with non-matching message interaction patterns in application interaction. In this chapter we describe the design and implementation of a protocol mediation component that has been applied in the interoperation between two heterogeneous logistic provider systems (using two different standards: RosettaNet and EDIFACT), for a specific freight forwarding task.

CURRENT SITUATION

Logistics is defined as the art and science of managing and controlling the flow of goods, energy, information and other resources like products, services and people from the source of production to the marketplace. As pointed out by Evans-Greenwood and Stason (2006) the current trend in logistics is to divide support between planning applications, which compute production plans overnight, and execution applications, which manage the flow of events in an operational environment. This disconnection forces users to deal with business exceptions (lost shipments, for example), manually resolving the problems by directly updating the execution and planning applications. However, this human-dependency problem can be ameliorated by

DOI: 10.4018/978-1-60566-066-0.ch004

using Web technology to create a heterogeneous composite application involving all participants in the process, providing a complete Third-Party Logistics solution, and giving users a single unified view into the logistics pipeline. This consolidated logistics solution greatly simplifies the task of identifying and correcting business exceptions (e.g., missing shipments or stock shortages) as they occur. Therefore, logistics management is a typical business problem where the use of a service oriented architecture is clearly suited.

Furthermore, Evans-Greenwood and Stason (2006) also talk about the possibility of combining multiple Third-Party Logistics solutions into a single heterogeneous virtual logistics network. With such a virtual network, each shipment is assigned a route dynamically assembled from one or more individual logistics providers, using dynamically created virtual supply chains. Most of these business functions are still manual and offline, but most of them can be automated with the use of service oriented architectures, as will be presented in this chapter. Obviously, the main advantages of using such solutions are the decreases in cost and speed in transactions, which influence in a better quality of the service provided to customers.

The main barrier to set up a business relationship with a company in the logistics domain is that it usually requires an initial large investment of time and money. This is ameliorated by the emergence of some industry standards like EDIFACT (EDIFACT), AnsiX12 (AnsiX12) or RosettaNet (RosettaNet), which ease the integration tasks between information systems that comply with them. However, given that these standards have some flexibility in what respects the content and sequencing of the messages that can be exchanged, the integration of systems is still time and effort consuming. Besides, there is sometimes a need to integrate systems that use different standards, what makes the integration task even more time and effort consuming.

This is the focus of one of the four case studies developed in the context of the EU project SWWS[1] (Semantic-Web enabled Web Services), a demonstrator of business-to-business integration in the logistics domain using Semantic Web Service technology. All the features of this demonstrator are described in detail by Preist and colleagues (2005), including aspects related to the discovery and selection of relevant services, their execution and the mediation between services following different protocols.

In this chapter we will focus on the last aspect (mediation) and more specifically on protocol mediation, which is concerned with the problem of non-matching message interaction patterns. We will describe the design and implementation of the protocol mediation component applied in this case study to show how to make logistic provider systems using two different standards (RosettaNet and EDIFACT) interoperate for a specific freight forwarding task.

The chapter is structured as follows. The rest of this section introduces a motivating example, focusing on the needs for protocol mediation, and gives some background on how the problem of mediation can be characterised in general and on the approaches for mediation proposed in the context of Semantic Web Service research. Section 2 summarises the protocol mediation approach followed for this case study and the main elements to be considered inside the approach. It also describes the ontology used for the description of the abstract and concrete protocols used by the entities involved in the message exchange. Section 3 provides an overview of the API of the protocol mediation component and gives details about how to configure it for deployment. Finally, section 4 gives some conclusions.

An Example in the Logistics Domain

Let us imagine that we have a manufacturing company in Bristol, UK, which needs to distribute goods internationally. The company outsources