Chapter 9

MATCHING AND MAPPING FOR SEMANTIC WEB PROCESSES

Tanveer Syeda-Mahmood[1], Richard Goodwin[2], Rama Akkiraju[2], Anca-Andreea Ivan[2]

[1]*IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120 -. stf@almaden.ibm.com*

[2]*IBM Watson Research Center, 19 Skyline Drive, Hawthorne, NY- rgoodwin,akkiraju,ivananca@us.ibm.com*

1. INTRODUCTION

A semantic revolution is happening in the world of enterprise information integration. This is a new and emerging field that blurs the boundaries between the traditional fields of business process integration, data warehousing and enterprise application integration. By information integration, we mean the process by which related items from disparate sources are integrated to achieve a stated purpose. For example, in data warehousing, data from two separate databases may need to be merged into a single database. This is particularly needed during mergers and acquisitions, where the respective company information from two separate databases may need to be merged into a single database. The terminology used to describe the same information in two disparate sources is hardly identical, subject to the vagaries of human use. Figure 9-1 illustrates two schemas from two databases that need to be reconciled during a data warehousing task. The two tables are called PurchaseOrder and POrder, respectively. They consist of 4 columns with names as shown. To properly merge such schemas, we need to reconcile the two terminologies and find their semantic relationships. Ordinarily, this is the job of a data warehousing specialist, who manually identifies the relationships using an application's user interface. Recent research is trying to make this process semi-automated by performing candidate matching between the names automatically, and having people verify the mappings.

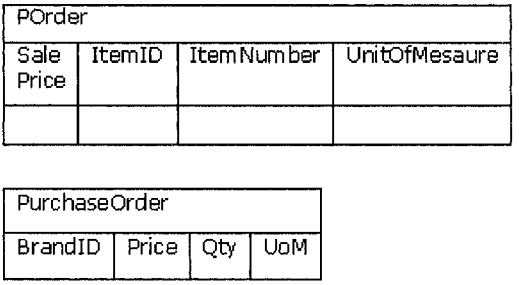

Figure 9-1. Illustration of schema matching in a data warehousing scenario.

Consider another scenario, now in the context of business process integration. Here a typical task may be a business flow that routes the data between suppliers and their associated applications. Typically, such flows are composed by business analysts who have limited programming skills, and work with user-interfaces that aid in the creation of business flows. They work with an abstraction of data being routed through schemas called business objects. Examples include generic business objects and application specific business objects made popular by CrossWorld (CrossWorld (2002)) a company that was later absorbed by IBM. These business objects are often encoded in XML syntax but are really structured data as illustrated in Figure 9-2. Here two business objects are depicted that come from two separate business applications, say, SAP (SAP (2005)) and Oracle e-Business Suite (Oracle (2002)) that both describe the concept 'Inventory'. The interface descriptions are shown here in the form of a tree for purpose of illustration here. In order to transform the output of one application into the next in a business flow, mapping of attributes from source to target schema is again needed. One such mapping is shown in Figure 9-2. The closely related terms shown by the arrows include some obvious cases such as terms (OrganizationID, OrgID) as well as non-obvious ones such as (InventoryType, StockType).

Our final example comes from the domain of web services. Service-oriented architecture is the latest trend in distributed computing where the need-to-know abstraction of object-oriented programming is again deployed. In service-oriented architecture, the capability of a code component anywhere on a network is described through an interface language called Web Service Definition Language (WSDL) (Chinnici, R., M. Gudgin, et al. (2003)). A WSDL describes a service as a collection of operational interfaces and their type specification, together with deployment