Towards an Approach for Enterprise Application Integration Based on Specialized Services and Business Objects

Zakaria Maamar\textsuperscript{1}, Youcef Baghdadi\textsuperscript{2}, Wathiq Mansoor\textsuperscript{3}
\textsuperscript{1}Zayed University, U.A.E, zakaria.maamar@zu.ac.ae
\textsuperscript{2}Sultan Qaboos University, Oman, ybaghdadi@squ.edu.om
\textsuperscript{3}American University in Dubai, U.A.E, wmansoor@aud.edu

Abstract

The recent development wave in the Enterprise Application Integration (EAI) domain promotes Service-Oriented Architecture (SOA) and one of its implementation technologies that is Web Services (WSs). Recently, Data Services (DSs) have risen as a new form of services that are specifically optimized for real-time data integration demands. Despite the role that WSs and DSs could both play in addressing enterprise application issues, there is still a lack of architectures and approaches to methodologically design solutions to these issues. In this paper we propose an architecture along with an approach that both make WSs and DSs work hand-in-hand during the exercise of integrating enterprise applications, whereby WSs and DSs are supported by reusable software components, which we refer to as Business Objects (BOs). In this architecture, BOs abstract the complexity and disparity of data resources and thus, constitute their unique access points; WSs interface changes in BOs; and DSs interface only retrieval of the status of BOs. WSs/DSs-BOs combination raises issues such as how WSs, DSs, and BOs are separated and empowered with appropriate mechanisms, how they collaborate, how WSs are differentiated from DSs, and how WSs and DSs access BOs in a controlled way.

Our proposed architecture combines WSs/DSs and BOs and spreads over four levels, which are denoted by (L1) enterprise application, (L2) service (specialized into Web service and Data service), (L3) business object, and (L4) data sources. The four levels are connected as follows: (I1: L1/L2) composition, (I2: L2/L3) matching, and (I3: L3/L4) access. Keywords: Application Integration, Web Service, Data Service, Business Object, Data Source.

1. Introduction

Enterprise Application Integration (EAI) domain is continuously fueled by challenges that today’s organizations face when they need a unique access to their scattered business-data resources. Different terminologies are used in this domain ranging from federation and interoperability to mediation and brokerage. Methods have used data-oriented integration, distributed object computing middleware such as COBRA [1], DCOM, or EJB, component-based solutions [2].

Despite the role that WSs and DSs could both play in addressing integration issues like making processes cross organization boundaries and agreeing on a certain data exchange format, there is still a lack of architectures and approaches to methodologically (and systematically) design solutions to these issues. Using WSs and DSs on a case-by-case basis yields brittle interfaces that would not properly support the requirements of running integration requests over disparate resources and abstract software components and data resources as well. In this paper we propose an approach, guided by an architecture, that makes WSs and DSs work hand-in-hand during the integration of enterprise applications, whereby WSs and DSs are supported by reusable software components, which we refer to as Business Objects (BOs). A BO abstracts disparate data sources and thus, provides a unique access point to applications. Examples of BOs could be bill, customer, and order, but in general, the profile and nature of an enterprise shapes the nature of the BOs. WSs interface allows making changes in BOs while DSs interface allows checking on the status of BOs. WSs/DSs-BOs combination raises issues such as how WSs, DSs, and BOs are separated and empowered with appropriate mechanisms, how they collaborate, how WSs are differentiated from DSs, and how WSs and DSs access BOs in a controlled way.

Our proposed architecture combines WSs/DSs and BOs and spreads over four levels, which are denoted by (L1) enterprise application, (L2) service (specialized into Web service and Data service), (L3) business object, and (L4) data sources. There are three mediations in this architecture; (I1: L1/L2) composition mediation, (I2: L2/L3) matching mediation, and (I3: L3/L4) access mediation. Data source is at the bottom representing raw data that enterprises own, whereas enterprise application is at the top representing software applications upon which enterprises rely for functioning.

The design and specification of the proposed four-level architecture are driven by the principles of separation of concerns [3] and abstraction. On the one hand, separation
of concerns stipulates that each level is responsible for particular duties that would facilitate the completion of the integration of enterprise applications. For instance, the business-object level accesses and integrates data sources on behalf of the service level. This level is the host of WSs and DSs that will be involved depending on the nature of this integration. On the other hand, abstraction stipulates that the business-object level hides the complexity and disparity of the data sources, and the service level itself offers the application level access to the business-object level. In other words, the approach’s potentials are as follows: (1) BOs abstract the complexity and disparity of data sources and thus, constitute the unique access points to these data resources; (2) WSs interface allows changes in BOs using the update operations that empower them; and (3) DSs interface allows checking on the status of BOs using the consultation operations that empower them.

The rest of this paper is organized as follows. Section 2 introduces an illustrative example. Section 3 details the proposed approach to the Enterprise Application Integration in terms of building blocks, architecture, and functioning. Section 4 presents some related works.

2. Running Example

Our running example is a simplified version of the "traditional" purchase-order scenario (Figure 1). A customer places an order for several products via Customer-App, which is afterwards submitted to CRM-App (with CRM standing for Customer Relationship Management). This application checks the customer's purchase history and sends Billing-App (via Customer-App) different details such as discount eligibility and reward points. Billing-App then prepares the customer's bill as per these details and sends it to Customer-App. Following the confirmation of the customer, the bill is sent to Payment-App for processing. In parallel, Customer-App sends Inv-Mgmt-App the approved order. Upon receipt, Inv-Mgmt-App checks the inventory of products in the stock. Depending on the inventory, the available products are directly shipped to the customer via Shipper-App, whereas the unavailable products are first, procured via Supplier-App and then, shipped like described before.

Figure 1 Illustration of purchase-order scenario

According to the aforementioned description, we consider the following mappings:

CRM-App are mapped onto a WS as it updates the customer's purchase order throughout the life cycle of completing this order;

Supplier-App are mapped onto a DS as it is used as intermediary between the company and suppliers;

And, item and customer are mapped onto BOs.

Overall, WSs change the status of BOs and DSs access BOs for data consultation and collection. This status change is reflected on the data sources (e.g., product database) that BOs interface.

3. The proposed approach to enterprise application integration

3.1. Architecture

Figure 2 shows the four-level architecture of the proposed enterprise-application-integration approach. This architecture is the host of several WSs, DSs, and BOs that all interact together through composition, matching, and access interfaces. We show later that BOs constitute a unique access point to disparate business data, which is the opposite of what some approaches and development tools do through DSs [4]. DSs are developed on a case-by-case basis, which limits their use opportunities, whereas a BO represents an identifiable business entity that is driven by the business domain not by specific needs. Thus, a BO that encapsulates business data and business logic can properly abstract the disparity and heterogeneity of the existing resources.

Figure 2 Proposed architecture for enterprise application integration

The interactions in the architecture of Figure 2 happen in a top-down way. They start from the enterprise-
application level, go through the service and business-object levels, and finally end at the data-source level. The enterprise-application level is about an enterprise's strategic goals for growth and expansion. To reach these goals, WSs, DSs, and BOs collaborate depending on the types and requirements of these goals. Collaboration could be vertical (between elements in adjacent levels) or horizontal (between elements at the same level). Both WSs and DSs have interfaces with BOs but for different purposes. Since WSs are active components, they make changes in BOs through update operations. Upon completion, BOs reflect these changes on the data sources to which they have access. Contrarily, DSs are passive components, and thus they only consult BOs to extract the data they need for processing.

Data-source, business-object, service, and enterprise-application levels are connected to each other through three types of mediation. We describe each mediation in terms of purpose and the levels it involves in the next section.

3.2 Implementation

The data service layer can be implemented by various technologies. It could be as simple as a basic web service. In this section it will be implemented using Oracle Data Integrator ODI [5]. Figure 3 shows that ODI can act as Data Service and Business Object layer that interacts with the database.

![Figure 3 Data Service Implementation](image)

The advantage of ODI is that it will serve as a data integration purpose as well. So even when there is a data grid of various formats ODI can accommodate them.

4. Conclusion

In this paper, a novel architecture has been proposed for Enterprise Application Integration based on specialized services and business objects. This architecture makes WSs and DSs work hand-in-hand during this integration and collaborate with BOs. In addition to the architecture an approach was discussed along the three actions it includes: (i) identify the components that should populate each level; and define how elements in a certain level should provide support to elements in other direct upper-levels (or how a certain level relies on a direct lower-level), (ii) identify interactions that could arise between elements in separate levels, and (iii) suggest mechanisms that should let these interactions take place. In the future work we will develop a BOs server that extends for instance the EJB or JDO of J2EE framework; in order consider the persistence of the business objects not only in relational data bases but on any kind of data sources.

5. References