Modeling Service Data Objects (SDOs) to the Entity-Relationship (ER) Model

Sikha Bagui, University of West Florida, USA
Richard Sweetman, University of West Florida, USA

ABSTRACT

In this paper the authors present a conceptual framework for translating Service Data Objects (SDOs) and XML's SDOs to the Entity Relational (ER) Model. With the increasing dependence on service oriented architectures and the increasing need for SDOs in service oriented architectures (SOA), it is important to have a good understanding of SDOs in terms of the ER model so that SDOs can be easily converted to the relational model. In this paper they show how common SDO constructs and XML's SDO constructs conceptually map to the ER model.

Keywords: Data Graphs, Entity-Relationship (ER) Modeling, Extensible Mark-Up Language (XML), Service Data Objects (SDO), Service Oriented Architectures (SOA), XML Schema Declaration (XSD)

1. INTRODUCTION

One of the major challenges facing businesses today is the need to respond quickly and efficiently in this global marketplace. To support flexible business collaboration, modern distributed applications are becoming component based and the concept of a service is becoming crucial. Recent years have witnessed a paradigm shift in distributed computing from middleware-oriented architectures (Baresi et al., 2003) and legacy systems (Li et al., 2007) to Web Services and Service Oriented Architectures (SOA) (Bouguettaya et al., 2006; Chen et al., 2006; Kim et al., 2007).

SOA provides fast, effective, inexpensive business process integration (Dan et al., 2007; Zimmerman et al., 2004), asset reuse and improved business agility (Anand et al., 2005; Chen & Huang, 2006; Gold et al., 2004; Malloy et al., 2006). SOA can be considered an architectural framework that supports integrating business tasks as linked services that can be accessed when needed (Chen & Huang, 2006). In an ideal SOA world, everything (Imen et al., 2007), including information (Dan et al., 2007), would be a service and all components and services would be loosely coupled, making SOA highly dynamic and flexible. All functions would be services (Chen & Huang, 2006), and services would be the building blocks of systems. Services would have well-defined
interfaces allowing consumers to know how to interact with them without having knowledge of the underlying platform (Erl, 2005). Services could be grouped together to form complex integrated processes and applications. Additions or modifications to the applications would be by adding/modifying services, independent of the application. Services could be added or removed dynamically at runtime (Baresi et al., 2003; Cotroneo et al., 2004). SOA will provide the basis for the next generation of distributed software systems (Mukhi et al., 2004).

SOA has three parts: service providers, service requestors and service registry (Baresi et al., 2003; Champion et al., 2002; Mukhi et al., 2004; Nielsen et al., 2006). Service providers are the owners that offer a service. They provide machine readable descriptions of their services and publish them in a publicly available registry (Baresi et al., 2003; Champion et al., 2002; Mukhi et al., 2004), which the service requestors can access when looking for services. Services and components interact by exchanging messages. And, one of the most important components of any service is the data.

SOA inherently deals with different kinds of data from heterogeneous data sources. One of the major challenges facing SOA in the last few years has been the need for an efficient data integration mechanism (Chen et al., 2006; Chen & Huang, 2006; Mukhi, 2004; Zhang et al., 2004) to efficiently handle data. Data relevant to today’s enterprise applications lives in a variety of information sources – relational databases, packaged applications and various home grown applications, and external Web Services (Borkar et al., 2006; Carey et al., 2003). In this paper our aim is to look at how data can be handled in the SOA environment with respect to service data objects (SDOs).

2. RELATED WORKS

Several works have looked at handling data in the SOA environment. In the SOA environment, data is typically from heterogeneous data sources. Data may be of poor quality; there may be performance and scalability issues, complexity in data manipulation, hence, to effectively handle data, many works have proposed building a data services layer (Cohen, 2007; Davydov, 2005; Goodson & Bloomberg, 2008; Henninger, 2005; Lawson, 2009).

Krizevnik and Juric (2010) presented an improved comprehensive architectural model, Master Data Management (MDM), for the storage and management of permanent data in SOA. The use of MDM for ensuring high data quality in SOA has also been addressed (Abai, 2006; Butler & Pollock, 2008; Kalogirou, 2007).

Krizevnik and Juric (2010) and several others (Brodsky & Stockton, 2005; Wright, 2009) proposed that all data transfer between data services and businesses goes in the form of Service Data Objects (SDOs). SDOs allow uniform data access and manipulation regardless of the data type, scalability, reliability, availability, and eliminate bottlenecks. This architecture also proposes advanced caching with the use of the data services layer as well as in-memory data grids.

Dan et al. (2007) treated information as a service. This means alignment of information sources. Dan et al. (2007) mentioned that all applications need to access persistent data stored in one or more data stores – databases of some sort. Dan et al. (2007) presented the benefits of using information as a service as: (i) loose coupling to data stores and data models, enabling aggregation of data from multiple sources, improving data quality; (ii) reuse and optimization of data access logic; (iii) support of data governance; (iv) separation of concerns – application developers no longer have to possess a sophisticated skill set in data management; and (v) flexible processes responding to new business needs; (vi) reuse of services across processes.

Though various studies have addressed the issue of efficient integration of data from heterogeneous data sources in various ways, the underlying concept for the efficient handling of data seems to be an MDM model of some sort with a data services layer and the use of SDOs to efficiently transfer data back and forth. SDOs
Related Content

Multi-Cloud Data Management Using Shamir’s Secret Sharing and Quantum Byzantine Agreement Schemes
Mohammed A. AlZain, Alice S. Li, Ben Soh and Eric Pardede (2016). Web-Based Services: Concepts, Methodologies, Tools, and Applications (pp. 1205-1222).
[Link](www.irma-international.org/chapter/multi-cloud-data-management-using-shamirs-secret-sharing-and-quantum-byzantine-agreement-schemes/140848/)

A Graphical User Interface (GUI) Testing Methodology
[Link](www.irma-international.org/chapter/graphical-user-interface-gui-testing/40426/)

Engineering Wireless Mobile Applications
[Link](www.irma-international.org/article/engineering-wireless-mobile-applications/2603/)

Towards the Adaptive Web Using Metadata Evolution
[Link](www.irma-international.org/chapter/towards-adaptive-web-using-metadata/21979/)

High Performance Scheduling Mechanism for Mobile Computing Based on Self-Ranking Algorithm
[Link](www.irma-international.org/article/high-performance-scheduling-mechanism-mobile/2607/)