RESEARCH ARTICLE OPEN ACCESS

A Web Service Based Design to Integrate a Web Portal and an ERP System

Mr. Manishkumar R. Solanki, Ms. Seema Shah
M.E.-INFT Student,Vidyalankar Institute of Technology, Wadala,Mumbai University.
Assistant Professor, INFT Department,Vidyalankar Institute of Technology,Wadala, Mumbai University.

Abstract
Today the ability to seamlessly exchange information between internal business processes, customers, and partners is vital for success. The organizations employ a variety of disparate applications that store and exchange data in dissimilar ways and therefore cannot "talk" to one another productively. Web services have evolved as a practical, cost-effective, deployment environment to realize dynamic and interoperable data integration between critical applications over operating system, platform, and language barriers that were previously impossible. This paper describes a service-oriented approach for the integration of a Web Portal and ERP application. We present a system architecture based on Web Service oriented framework in which both the applications are loosely connected through software components.
Keywords—ERP,SOA, SOAP, VERP, VIOLA, UDDI, Web Portal,WSDL

I. Introduction

Enterprises aretypically comprised of hundreds applications that are custom-built, acquired from a third-party, part of a legacy system, or a combination of these types, operating in multiple tiers and running on different operating systems and platforms. [1] Data can be found throughout the enterprise, in multiple disparate systems and in many different formats. Data is scattered everywhere—on the mainframe, in databases, in obscure legacy systems, in spreadsheets on desktops, in enterprise resource planning (ERP) applications, on message queues, in flat files. [2]

Data integration allows organizations to access all their fragmented data, create an accurate and consistent view of their core information assets, and easily leverage these assets across the enterprise to drive business decisions and operations. Organizations are using data integration in many different ways to drive business value. They are using data integration to migrate data into new applications, or implement master data management. They are also using data integration to synchronize data across operational processes and systems, and to create flexible, reusable data services.

The goal of this paper is to propose a system design for the data integration through Web Services approach. The remainder of the paper is organized as follows: Section II discusses various issues of data integration. Section III presents the concept of Web Service, Section IV suggests the design architecture, and Section V gives brief idea of implementation and at the last Section VI concludes the paper.

II. Issues related to data Integration

Since a typical IT organization usually has a collection of isolated, incompatible, disconnected applications and databases [3], it is a prime need for data integration to give a user the unified access of the overall system. At the other end, companies deploy their applications on different database products i.e. Oracle, SQL Server, Sybase, MySQL, etc. rather than one. it means that different data interfaces should be considered for data integration. But even if databases federated nicely, which they don’t, there is still a mass of data locked in flat files of various sorts, or content management systems or document management systems. As the new software applications are being developed in very much rapid manner, the biggest challenge is to integrate the data between the legacy systems and these new applications.

In the past, companies had to bet the business on CORBA, JAVA RMI, distributed small talk, or DCOM to create a SOA. They yielded systems where the coupling between various components in a system is too tight to be effective for low-overhead, ubiquitous B2B e-business over the Internet. For example, DCOM clients access remote COM types using tightly coupled RPC calls. CORBA requires the use of tightly coupled protocol referred to as Internet Inter-ORB Protocol (IIOP), to activate remote types. Enterprise JavaBeans (EJBs) requires a Remote Method Invocation (RMI) Protocol and by and large a specific language (Java). [4] Thus each of these remote invocation architectures needs proprietary protocols, which typically require a tight connection to the remote source. These approaches require too much agreement and shared context among business systems from different organizations to be reliable for open, low-overhead B2B e-business. [5]The risks and costs of standardizing on one outweighed the potential benefits of SOA. This accounted for the low adoption rate.

Existing mission critical applications can be wrapped in Web services interfaces and then accessed from other applications or Web browsers. This enables businesses to create business services out of existing systems, and rapidly implement and integrate new functionality. Using Web services, companies can begin leveraging existing investments and incrementally adding new functionality. [6]

III. The Concept of Web Services

The Service Oriented Architecture (SOA) has facilitated to develop discrete business functions or processes as independent components with standard interfaces which can be accessed by other applications, services, or business processes, regardless of the platform or programming languages. Web services are offering a technology that is rich and flexible enough to make SOAs a reality. [7]

"A Web Service is a self-contained, modular, and dynamic application which is described, published, searched and invoked over the network to create procedures, business processes and supply chains, can be local, distributed or web based and uses open standards i.e. HTTP and XML and open protocols i.e. HTTP, SMTP, BEEP, TCP/IP, etc. to support heterogeneity across the platform and language boundaries."

A. **Architecture:**

A typical Web Services architecture consists of three elements as shown in Fig.1:

- A Service Provider develops services which he wants to provide to the outside world and publishes this services to the UDDI registry
- A Service Requester or The Client looks up the services at the UDDI registry and binds with the Service Provider for the services which he wants to access.
- A UDDI registry is a central repository which maintains the registry of published services.

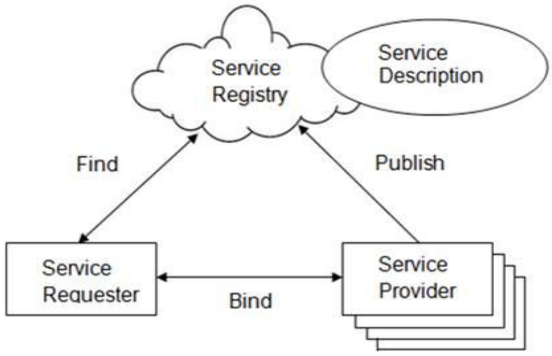

Fig. 1 A Web Service Architecture

B. **Technologies:**

Web Services are based upon three main technologies WSDL (Web Service Description Language), UDDI (Universal Description and Discovery Integration) and SOAP(Simple Object Access Protocol).

WSDL:

The Web Services Description Language (WSDL) is written with an XML grammar for describing Web services as a set of network endpoints that operate on messages. A WSDL service description contains an abstract definition for a set of operations and messages, a concrete protocol binding for these operations and messages, and a network endpoint specification for the binding. [8] Three major elements of WSDL that can be defined separately and they are:

- Types (data types of parameters to be passed to and to be retrieved from the Web Service)
- Operations (an abstract description of an action supported by the service)
- Binding (a concrete protocol and data format specification for a particular port type.)

The service requester should know the above elements to invoke a particular Web Service while writing the Web Service client code.

WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the exchanging of messages between applications. [8]

UDDI:

It specifies registry of Web services. It defines an electronic business registry where businesses can describe their business and register their Web services as well as discover and integrate with other businesses that offer Web services. [9]Service providers can dynamically publish information and services, and service requesters can search through the registry to find personal information about a service provider as well as technical information about its services. UDDI maintains four basic information about the Web Service i.e. businessEntity (information about the party who publishes web service), businessService (description of technical services), bindingTemplate (provides the details of how and where the service is accessed) and tModels (abstract service protocols that describe a particular Web Service's behavior.) [10]

SOAP:

SOAP provides a simple and lightweight protocol for exchanging XML data over the Web. SOAP enables client applications to easily connect to remote services and invoke remote methods. SOAP messages are written entirely in XML and are therefore uniquely platform- and language-independent compared to the protocols used by CORBA, DCOM, and Java RMI, frameworks.

SOAP consists of three parts:

- The SOAP envelope (defines an overall framework for expressing what is in a message; who should deal with it, and whether it is optional or mandatory)
- The SOAP encoding rules (defines a serialization mechanism that can be used to exchange instances of application-defined data types)
- The SOAP RPC representation (defines a convention that can be used to represent remote procedure calls and responses)

The next section discusses the proposed system design to implement the data integration of a Web Portal and an ERP system.

IV. System Design

Vidyalankar Institute of Technology(VIT), Wadala (E),Mumbai has implemented an Academic Social Networking Portal, which provides with the right mixture of academic services, entertainment and socialization. These features were instrumental for the foundation of VIOLA which stands for 'Vidyalankar Online Applications'. It provides an interactive and user-friendly medium for all its users from anywhere round the world.

VIT is also running an ERP System named VERP which integrates and automates the data of Accounts, Admission, Exam, Library, Human Resources, Administration, Management, Payroll and other departments on a single computer system provide a unified central access of information of entire institute.

Presently, both these systems are running independently. But some data is commonly needed in both the systems. As a result when common data changes we have to update it in both the systems manually which is a cumbersome process. Eventually, some of the data across systems became inconsistent. When institute noticed the resulting double data entry, inconsistent data, and data isolation problems, the management decided to find ways to integrate the systems.

Requirements:

- The profile information of the user in VIOLA should be integrated to VERP
- Subject details from VERP should be imported to VIOLA for transcript generation and to fill faculty feedback forms
- The student details from VERP should be integrated to VIOLA to generate the Bonafide Certificate and to send it to the student through an email.
- The Company and the student details from VERP should be integrated to VIOLA so that the Training and Placement Officer can conduct the campus interview online.

SOAP based Web Service provides solution for this problem by creating web services i.e. VIOLA WS and VERP WS so that each system can invoke its relevant web service to integrate its data with other system to keep both the systems consistent.

The data integration between VIOLA and VERP systems will be taking place by developing Web Services. How these two systems will communicate is depicted in the architectural diagram shown in Fig. 2. Explanation of this diagram is given below:

Steps of Communication:

1. The first task is to develop relevant web services i.e. VIOLA WS (Web Services) and VERP WS (Web Services) which will be deployed at VIOLA and VERP systems respectively.
2. We have to create the WSDL (Web Service Description Language) files and these files have to be published (registered) with UDDI (Universal Description and Discovery Interaction) residing on the central server which is Linux Cent OS Server. [As shown with label no. 1(Publish) in the diagram]
3. The client (VIOLA or VERP) will find the WSDL files on the central server and has to understand the request and response parameters to invoke a particular web service. [As shown with label no.2 (Find)in the diagram]
4. The client has to call the desired web service which is called as a bind process.
   [As shown with label no.3 (Bind) in the diagram]

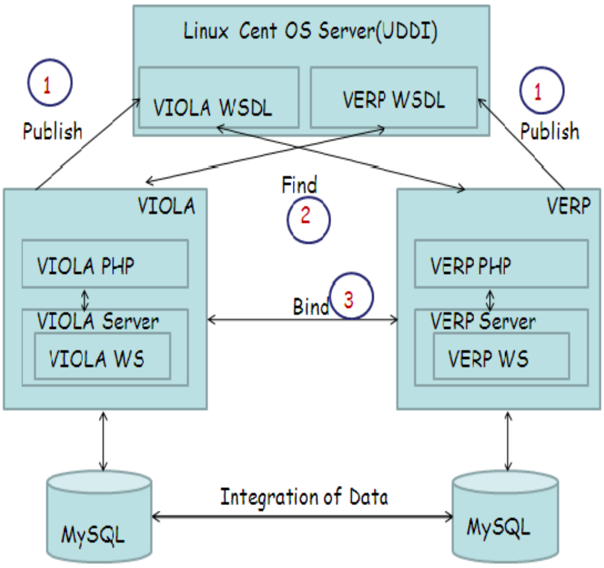

Fig. 2 VIOLA-VERP Integration Architecture Diagram

V. Plan of Implementation

VIOLA and VERP are developed in PHP while Web Services will be developed in Java. So a PHP based application will invoke the Java based web services. The WSDL files will be generated with the help of Apache Axis2 Toolkit. These files will be deployed at the Linux CentOs Server. The Business Logic of web services will be written in Java and after compilation, .aar file will be generated to deploy it inside the services directory of Axis2 Toolkit. Axis2 Toolkit will be running inside the Tomcat application server which is providing a run time environment for Java based web applications. Finally, both VIOLA and VERP will generate the relevant stub classes in PHP to invoke the web services to pass parameters and to retrieve the return values (if necessary).

VI. Conclusion

The Web Services enable a distributed environment in which any number of applications, or application components, can interoperate seamlessly among and between organizations in a platform-neutral, language-neutral fashion. The data integration between the present running systems and legacy systems can be achieved by the web services in cost effective manner. To examine the said advantages of Web Services we have proposed the design to integrate a web portal (VIOLA) and an ERP system (VERP).

References

[1] http://www.eaipatterns.com/Chapter1.html
[2] Enterprise Data Integration, Maximizing the Business Value of Your Enterprise Data - http://www.integralsolutions.pl/is/images/artykuly/enterprise_data_integration.pdf
[3] http://www.dadbm.com/2011/11/enterprise-data-integration-issues-due-to-software-complexity/
[4] http://www.roseindia.net/webservices/webservices.shtml
[5] http://www.ibm.com/developerworks/webservices/library/w-ovr/
[6] Enterprise Integration: The Essential Guide to Integration Solutions,By Beth Gold-Bernstein, William Ruh,Published Jul 16, 2004 by Addison-Wesley Professional. Copyright 2005, Edition:1st,Book,ISBN-10: 0-321-22390-X,ISBN-13: 978-0-321-22390-6
[7] Kevin J. Ma,Web Services: What's Real and What's Not?, IEEE Computer Society,March-April 2005,pp 14-21
[8] http://www.w3.org/TR/wsdl
[9] Jaideep Roy and AnupamaRamanujan," Understanding Web Services", IEEE computer Society,November-December 2001, pp 69-73
[10] https://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm