Migrating Legacy Information Systems to Web Services Architecture

Shing-Han Li, Tatung University, Taiwan
Shi-Ming Huang, National Chung Cheng University, Taiwan
David C. Yen, Miami University, USA
Cheng-Chun Chang, National Chung Cheng University, Taiwan

ABSTRACT

Web services-based systems with Service-Oriented Architecture (SOA) are widely accepted as one of the possible solutions for an enterprise information system to retain/keep its old legacy systems. Using this aforementioned architecture, enterprise information systems tend to be more flexible and agile to fit into the capricious business environment, and thus, will be easier to be integrated with other additional applications. The purpose of this manuscript is to propose a migrating solution to convert the architecture of the legacy system into SOA with a systematic approach. This proposed methodology is different from other traditional object-oriented approaches, which migrates the system to have a services-oriented focus without incorporating general object-oriented (OO) or functional oriented feature. In this study, a case study and information capacity theory were employed to verify(validate) that this approach is indeed an effective and a practicable one.

Keywords: legacy system; information system reengineering (ISR); service component; service-oriented architecture (SOA); Web services

INTRODUCTION

Due to the dynamic advancement of information technology (IT), the life cycle of the information system (IS) is greatly reduced to a certain extent. Generally speaking, the traditional legacy information systems possess such undesirable characteristics as latency of information, poor reach, inflexibility, and higher cost of maintenance. Furthermore, the traditional system architectures such as centralized and client/server are frequently incompatible with the requirements and specifications which exist in today’s business environment. To be more specific, the legacy information systems have these aforementioned shortcomings,
which have prevented the businesses and/or organizations to react/respond dynamically to the rapid challenges as they should. Consequently, enterprises have a strong need to utilize the technology of information system reengineering (ISR) to preserve the value of their legacy systems.

In this situation, enterprises or software companies are always in a dilemma of redeveloping/redesigning their legacy systems to include the newer Web services components (Bouguettaya, Malik, Rezgui, & Korff, 2006; Chen, Zhou, & Zhang, 2006; Kim, Sengupta, Fox, & Dalkilic, 2007). Discarding and redeveloping the existing systems not only wastes the money allocated for software investments, but also causes organizations to lose competitive advantages to meet numerous unanticipated contingencies and/or uncertainties. Based on prior study (Ommering, 2005), the system migration will be one of the best ways to reengineer a legacy system. Traditionally, there are two approaches available to migrate the legacy system to the Web services architecture (Vanston, 2005). The first approach is the legacy externalization approach. This approach is usually the main alternative available on the current market. It generally uses strategic or pointed forms, along with new types of interface display, to develop the integrated products (such as “Web Scraping”). The other approach is the component encapsulation approach. This is another viable alternative to utilize the component standard technology like Common Object Request Broker Architecture (CORBA) (OMG, 1995; Vinoski, 1997), Component Object Model (COM) (Microsoft, 2007), or Enterprise Java Beans (EJB) (Sun, 2007) to encapsulate the legacy system into the components, and then translate them into a Web Services standard. Ultimately, this second approach is migrated to the component-based and transaction-oriented framework (such as IBM WebSphere and BEA WebLogic) (Liu, Fekete, & Gorton, 2005; Waguespack & Schiano, 2004). Both of the aforementioned approaches may not be a bad way for the legacy system to migrate into the equivalent Web services standards. However, they normally utilize the hard-cording technique to implement the interface with the corresponding standard (Brereton & Budgen, 2000; Kwan & Li, 1999; McArthur, Saiedian, & Zand, 2002). Being a traditional structure program, the system normally has a shorter life cycle and lacks scalability, feasibility, and reusability. Further, it would be much more difficult to maintain in the future. On the other hand, if a company is applying the component encapsulation approach without incorporating appropriate component migrating methods, the system still has these aforementioned shortcomings (Rahayu, Chang, Dillon, & Taniar, 2000). Unfortunately, most alternatives adopted now by enterprises and/or businesses do not use the proper component migrating method.

Many related studies (Erickson, Lyytinen, & Siau, 2005; Fong, Karlapalem, Li, & Kwan, 1999; Gall, Klosch, & Mittermeir, 1995; Kwan & Li, 1999; Sang, Follen, Kim, & Lopez, 2002) have presented methods that can be utilized to systematically reengineer the legacy system into the Object-Oriented (OO) or the distributed system. However, the Web Services architecture by nature is different from a general distributed system. The core concept of Web Service is a Service-Oriented Architecture (SOA) (Huang, Hung, Yen, Li, & Wu, 2006; Stal, 2002). In the SOA environment, resources in a network are made available as an independent service that can be accessed
Related Content

The Expert's Opinion
www.igi-global.com/article/expert-opinion/51116?camid=4v1a

A Scalable Middleware for Web Databases
www.igi-global.com/article/scalable-middleware-web-databases/3361?camid=4v1a

INDUSTRY AND PRACTICE: An Empirical Investigation of the Effectiveness of Object-Oriented Database Design
www.igi-global.com/article/industry-practice-empirical-investigation-effectiveness/51141?camid=4v1a
Entity-Relationship Modeling and Normalization Errors
www.igi-global.com/article/entity-relationship-modeling-normalization-errors/51172?camid=4v1a