Available online at www.sciencedirect.com

SCIENCE DIRECT®

Decision Support Systems 40 (2005) 129–141

Decision Support Systems

www.elsevier.com/locate/dsw

Optimal location and pricing of Web services intermediary

Qian Candy Tang, Hsing Kenneth Cheng*

Department of Decision and Information Sciences, Warrington College of Business Administration, University of Florida, P.O. Box 117169, Stuzin 351, Gainesville, FL 32611-7169, USA

Available online 18 May 2004

Abstract

The Web services technology allows for the distribution and integration of loosely coupled software components over the Internet. This paper studies the optimal pricing and location strategy of a Web service intermediary (WSI), which offers a *time-sensitive composite* Web service. We first derive the optimal solution in a linear city model and then extend the analyses to the more general unit circle model. Our analyses show that that the optimal strategy is determined by delay cost, integration cost, and prices of the constituent Web services. We find that the WSI is optimally located between the Web service providers and charges a penetration price if the delay cost is low. In addition, there could be multiple optimal locations for the WSI if the Web service providers are far away from each other.

Keywords: Web services intermediary; Spatial model; Pricing; Location model

1. Introduction

The software industry has long been a battlefield where firms try to win customers via proprietary platforms and technologies. Web services, a new computing paradigm, for the first time is gaining endorsement simultaneously from major software developers like IBM, Sun, and Microsoft, who traditionally compete with each other with their own technologies. According to the Stencil Group, Web services are "loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols" (www.stencilgroup.com), see also Refs [3,5]. From the technical perspective, Web Services represent a collection of standard protocols including XML, SOAP, WSDL, and UDDI for the creation, distribution, discovery, and integration of semantic software components that encapsulate business functionalities. From the business perspective, Web services enable just-in-time software service provisioning through the integration of loosely coupled software components. Central to the Web services architecture are the concepts of software as service and platform independence.

As opposed to packaged monolithic applications that have to be developed or licensed, Web services encapsulate specific business functionalities that can be "rented" over the Internet. Web services decompose business processes into granular components and thus allow customers to select the services on an as-

* Corresponding author. Tel.: +1-352-392-7068; fax: +1-352-392-5438.

E-mail addresses: qtang@ufl.edu (Q.C. Tang), hkcheng@ufl.edu (H.K. Cheng).

0167-9236/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2004.04.007

needed basis. The service-oriented architecture also opens new business opportunities for firms by allowing them to sell their software components as Web services over the Internet. For example, CitiBank developed CitiConnect, a payment-processing service that can be plugged into other company's transaction process [6]. The spectrum of Web services spans from personal services such as stock quote, messaging services to enterprise-centric services such as call center control, payroll management, shipping and logistics, and so on.

Another key feature of the Web services is the openness of technology. The stack of Web services core protocols includes XML, WSDL, SOAP, and UDDI. The XML (extensible Markup Language) protocol allows self-describing data to be exchanged independent of platform and language. The XML-based messaging protocol SOAP (Service-Oriented Architecture Protocol) supports the invocation of components similar to RPC (Remote Procedure Call) on existing Internet network. The WSDL (Web Services Description Language) builds on XML and describes what the Web service does, where it is located, and how to communicate with it. The UDDI (Universal Description, Discovery and Integration) represents a set of protocols for the description, registration, dynamic lookup, and integration of software components. In essence, the Web services architecture provides a platform-, language-, and vendor-neutral framework for the interaction and integration of software components via standard networking technologies.

The modularity and platform independence of Web services will greatly affect software development and deployment. Internally, the Web services technology changes the fundamental cost structure of software development and enterprise application integration. By leveraging existing systems and outsourcing standard modules, a firm can decrease software development cost and duration dramatically. With interoperable software components that encapsulate business functionalities available over the Internet, firms are endowed with the flexibility of choosing the best-of-breed software components and pay for the services on an as-needed basis. Externally, Business-to-Business (B2B) integration or collaboration is made more cost-efficient because the firms no longer have to set up a separate integration project with each business partner. As a result, business alliances can be created and decoupled on the fly and firms can dynamically lookup, bind, and consume Web services over the Internet.

Furthermore, the flexibility, reusability, and interoperability of the Web services paradigm help create a new business model of Web services intermediary (WSI). A WSI provides value-added services including directory and search engine, auditing, quality-of-service (QoS) assurance, as well as the integration or orchestration of Web services [1,4]. For example, salcentral.com, which originally called itself "the Napster of Web services", provides a Web services search engine and tools to develop and integrate Web services. Both the consumers and producers of the Web services benefit from the maintenance service provided by the WSI, which hosts the latest version of the Web services and offers tools to manage and control them.

While the Web services paradigm is a promising solution to bridge the platform discrepancy and geographical distance between software components, some constraints may hinder the widespread adoption of Web services and its service-oriented framework. For example, the performance of a service-oriented architecture is restricted by the computing power of local servers and the robustness and capabilities of the underlying network through which the Web services are distributed. Response time is especially important for time-sensitive applications such as stock quote and instant messaging. With computing power doubled every 18 months, according to the Moore's Law, the ability to increase the processing power of local servers at decreasing cost has become the norm. In contrast, the network infrastructure is less scalable and network topology is even more inflexible to change. Johansson [7] suggests that the network latency, which is directly related to the physical distance between two entities on the network, constitutes a rather salient proportion of response time in high bandwidth network. Therefore, the success of a Web services-oriented architecture requires prudent planning of the Web services distribution network.

In this paper, we focus on the optimal strategy of a Web service intermediary (WSI), which offers a new *time-sensitive composite* Web service by integrating two complementary Web service components. At the same time, the constituent Web service components are offered by two independent service providers.