A Web-Services-Based Personal Information Integration Framework

Sumi Helal, University of Florida, USA  
Jingting Lu, University of Florida, USA  
Erwin Jansen, University of Florida, USA

ABSTRACT

Managing fast-growing personal Web information is a time-consuming, laborious task that affects people’s daily lives. A framework is highly desired that will allow people to collect and fuse personal information without dealing with complex emerging technologies. In this paper, we present our E-Service-Based Information Fusion Framework for end users. This framework enables end users to collect scattered information from diverse autonomous sources and to transparently create a repeatable process by which newer instances of the same information can be obtained in the future. By exploiting this framework, users do not need to repeat the manual information-gathering task over and over again. We present our framework and provide some implementation details.

Keywords: information fusion; personal information systems; Web Services

INTRODUCTION

With the fast pace in which we live today and with the information society we are becoming, people have more and more personal information on the Web concerning many aspects of their lives. This includes online brokering accounts, bank accounts, credit cards, frequent flyer programs, to mention just a few. People often spend a substantial amount of time gathering and managing such information every time this information or a summary of it is needed. Usually, such information is scattered across different business sites. For example, many people have three or four bank accounts, more than one credit card, and several airline frequent flyer accounts. For regular online services, an end user needs to go to each individual Web site to authenticate and manually fill in the necessary details in order to invoke a service and get the information (i.e., balance a checking account). This is a time-consuming process that will be repeated every time the end user seeks a more up-to-date version of the information.

An alternative way to access information from different sources is by using the emerging e-service technology. An e-service is viewed as “any service or functionality that can be accessed by a business or a consumer programatically on the Internet, using standard representation and protocols” (Krithivasan & Helal, 1999).
2001). It can greatly improve the efficiency of invoking and integrating services. On the other hand, it involves fairly professional and complicated processes for end users, requiring them to have significant knowledge of e-service-related specifications. Even for e-service specialists, it is their responsibility to modify the service requests correspondingly, if a particular service interface is changed later on, or to deal with status query directly for checking back the execution status of a long-running service. Regardless of its efficiency, the plain e-service framework is clearly a complex and inconvenient way of invoking services and gathering information.

The major goal of our E-Service-Based User-Level Information Fusion Framework is to tame this complexity and simplify the use model of the emerging e-service standards, so that services can be used transparently by non-expert users. This can be done through a framework that involves a methodology at the e-service provider side and a carefully designed, user-friendly interface that facilitates the access and integration of information on the Internet at the end user side.

**Motivating Scenario**

Let’s meet a software engineer, Kin lee, who has three bank accounts in three different online banks: Bank One, Bank of America, and Hong Kong Bank. Every time he checks his three accounts’ balances, Kin spends a good deal of time going into each individual bank’s Web site and filling out multiple forms to finally get the bank balances. Fig. 1.1 depicts this balance-querying scenario.

Before he can receive the balance at Hong Kong Bank, Kin needs to make $n$ interactions with the Web site to input his context, such as account, password, and so forth. Similarly, $m$ interactions are needed at Bank of America, and $k$ interactions at Bank One. Hence, Kin needs to do a total of $(n + m + k)$ interactions in order to retrieve all three balances and finally calculate net worth. Apparently, he has to repeat the $(n + m + k)$ interactions every time he wants to check the balances, as well as memorize and put the accounts and passwords information somewhere safe. This process could turn into a nightmare and a waste of time for future Web users. This can only exacerbate the more Web users who opt to the electronic statement option of the various businesses.

We know e-services can be invoked in a standard messaging through the Internet, facilitating different businesses to build their applications on different systems and technologies they prefer. Thus, Hong Kong Bank, for instance, may implement the balance service as an e-service and publish it in a public e-service repository, allowing more potential users (businesses and individuals) to use the service easily. The process that Kin will follow, if he wants to exploit the e-service version of balance service is as follows: He first needs to refer to a public e-service repository to find the Hong Kong Bank’s e-balance service description file, then he generates the appropriate request message containing his context, and finally, he invokes the e-service to get the balance result. All exchanged messages should be based on some standard message format. If the service interface is changed later on, Kin will get an error message when he tries to invoke the service again. Thus, he must retrieve the service’s new description file in order to change the service request accordingly. Obviously, this process is more suitable for businesses, since they can invest much on implementing the corresponding software framework, whereas it is not necessary and practical for Kin to master the e-service-related emerging technologies. A user-level e-service interface would be very beneficial here.

Fig. 1.2 depicts our framework. It shows a more efficient process to query three different banks for three different balances from an Information Integration module without manually inputting any data by Kin. This process involves $(n + m + k)$ interactions only the first time Kin queries the three banks. During these interactions, Kin transparently and reflexively extracts the user-service-based information-gathering process, such as the e-service standard request message. In the future, Kin will not visit each bank’s Web site and repeat the same interaction steps. Instead, Kin will use
Rapid Development of Adaptable Situation-Aware Service-based Systems
S.S. Yau, S. Mukhopadhyay, H. Davulcu, D. Huang, R. Bharadwaj and K. Shenai
(2010). Web Services Research for Emerging Applications: Discoveries and Trends
(pp. 104-139).
www.igi-global.com/chapter/rapid-development-adaptable-situation-aware/41520?camid=4v1a