An Novel Web Mashup Discovery Framework

Ying Yin, Xizhe Zhang*, Honghong Xu, Bin Zhang
College of computer science and engineering, Northeastern University
*Corresponding author: zhangxizhe@mail.neu.edu.cn
Shenyang, Liaoning, 110819, China

Abstract

Mashup as a pattern for end-user applications is in the ascendant with the Web applications are widespread emergence. How to search for the novel and the suitable application of mashup becomes an urgent problem to solve. This paper presents a new method based on frequent mining of Mashup application. It is based on frequent analysis in order to find the core data set. To get novel mashup application services on the basis the core date sets. Firstly it is based on its structural parameter information of the interface describing documents of the web application, to find core data sets for frequent related web application. Then according to the output of application in heterosexual to filter the related Web application .Finally obtained Mashup Web application sets can be obtained.

Keywords: Mashup; web services; frequent mining

1. INTRODUCTION

Mashup (Braga, 2008) is a web application pattern for the end user. It creates a new web application by aggregating the existing web content or service, offering aggregation service for users' individual requirements to satisfy the diversified demands. Mashup as a kind of user personalization resource polymerization technology has been widely used. Aiming at web data source, opening API, Web Services, etc. the existing development tools, such as IBM QED Wiki, Yahoo! Pipes, Microsoft Popfly provide programmable extension. Encouraging the third-party developers and users to reorganize the existing resources with innovative ways and build Mashup application service.

The Mashup application processes, integrates, and further utilizes the existing information from multiple data sources (public APIs, XML/RSS/Atom feeds, web services, HTML) to generate greater value. The key of Mashup is innovation, that is, how to integrate multiple application services into new applications, to achieve the effect of 1+1>2. At the same time, with the development of web 2.0, Mashup applications are growing rapidly. According to the records of programableweb.com, from May 2007 to August 2010, the API exposed in network increased from 431 to 2306, however, the number of Mashup applications increased from 1900 to 5632. Thus, how to discover novel and polymerizable Mashup patterns from these massive Web application services becomes one of the most urgent problems to be solved.

Currently, the research work of Mashup mainly includes Mashup data conversion and integration, Mashup system design, Mashup platform and tools etc. ref first pointed out that Mashup is to integrate multiple information sources to create a new Website or network applications. Based on this, ref mentioned the basic architecture of Mashup on semantic Web Services, which is divided into service providing layer, semantic adding layer, Mashup layer and service consuming layer. Aiming to the problem of web service Mashup, ref references WSDL in SOA and ontologies used in automatic web service composition, and proposes a Mashup framework based on RESTful Web services. There are also some Mashup system designers used inquiry mode to integrate data from multiple sources. In order to integrate multiple Feed sources that change with time, J. Tatemura et al. ref proposed to use a continuous query on these Feed sources to capture timely and integrate results. ref regarded Mashup as a dynamic network composed of a set of interactive components, which was regarded as a relational database. The above work made in-depth study for the Mashup construction and framework. However, how to choose the right service for the integration and how to find new, available service model, which are still unresolved problems.

Observing the current Mashup applications, we can see that most of the applications are built around some core data, such as geographic applications centered on location data, shopping applications centered on product data, news applications focused on Interest, etc. As shown in Table 1. From these applications, we can see that it's one of the important factors to Mashup around the core data which is suitable for the application in the construction. Based on such an idea, this paper proposed a new Mashup application discovery method in order to find relevant applications for fusion potentially. The algorithm based on frequent mining to find core application data sets in the massive application sets. On the basis of it, we perform the Mashup integration.

DOI: 10.29268/iciot.2016.0009

In the algorithm, the web application interface is extracted firstly, and then, preforming parametric modeling and matching to it, finally, forming candidate parameter sets. Performing frequent analysis for parameter set space based on frequent analysis method, and get applied theme set currently. Performing de-overlaying operation for the theme to get Mashup core data, finally, according to the functional dissimilarity of the application service, doing novelty filtration for the relevant applications of core-items, and get the Mushup results. The method proposed in this paper takes into account the construction features of the mashup application, and the results can provide intelligent support for users to build markup applications.

This paper is organized as follows: section 1 mentioned the ideas and framework of Mashup discovery; Section 2 proposed the detailed steps of the framework; Section 3 summarize the paper.

2. WEB MUSHUP DISCOVERY FRAMEWORK

Figure 1 shows the whole Web mushup discovery framework. The framework includes several steps: (1) Web Application Interface Description Document; (2) Mining the core data; (3) Web Application Frequent Sets; (4) Web Mashup Application Sets. For each step, we will describe the process in details.

2.1 Web application preprocess

The objects processed by Mashup application is web services, API, RSS, XML documents, etc. Before mining, it needs to be preprocessed so that it can be excavated and analyzed in a unified format.

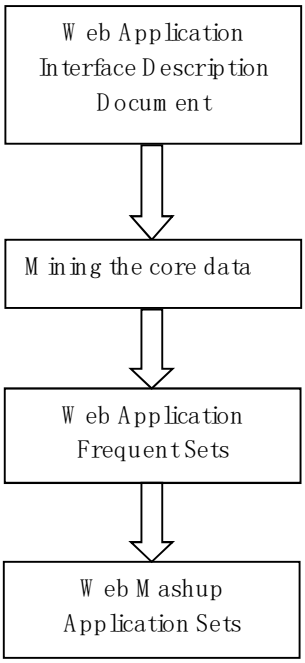

Fig. 1. Web Mushup Discovery Framework

For web services, we do the following processes: parse the Web service WSDL document, extract the Web service name, operation name, As well as the operation corresponding input parameters (the parameter name, type), output parameters (the parameter name, type). Then, a series of names and types of information extracted from the WSDL document are preprocessed, so that unifying the format to perform frequent mining and analysis.

2.2 Mining the core data

Mashup applications are typically organized around some core data, the associated APIs provide extended or supplemental applications for these applications' core data, and they are more or less associated with the core data. Such as, in example 1, the core data is retail information of the home game console Wii. In example 2, the core data is the news information of Japanese nuclear crisis. In example 2, the core data is the information of stars. Therefore, finding appropriate, innovative core data for web applications is one of the keys to build Mashup applications successfully.

Example 1. WiiSeeker : The application helps users find retail information about the home game console Wii, such as retailer, location, date of manufacture, price and local Ebay online auctions, and then displays these on the map. Associated APIs are: Amazon eCommerce、 eBay、 Google Maps. As shown in Fig. 2 (a).

(a)WiiSeeker

(b)RDTN

Figure 2. Typical Web Mashup Examples

Example 2. RDTN: The application provides a diffusion map of the Japanese nuclear crisis. By integrating data from multiple sources, such as facebook, twitter and

web news, and it will displaythe extent and scope of the leakon the map, then providing a fair analysis of nuclear leaks.Associated APIs are: Facebook Social Plugins, Google Maps, Pachube, Twitter. As shown in Fig. 2 (b).

2.3 mining frequent item sets

For the parameters information of analysis and extract apply the Apriori algorithm to calculate the input data and output data frequently sets. The set of all the operations of the transaction and the frequent item sets containing the same input parameter constitute an input frequent set; Similarly, all the operations in the aggregate transactional libraries contain frequent operations with the same output parameter, which constitute the operation output frequent set. Given the definition of the term set, using the Apriori algorithm to get the results. The operation input (output) frequent set contains one or more frequent items, and They occur at the same time in the operating transaction database is not less than minsup. That is to say, there is a certain number of transactions in the operation transaction that contains all the items of frequent set. The parameter frequent set is also the description tag of the next step related aggregation, which is used to distinguish the features of different Mashup Web correlation sets.

2.4 Web Mashup Application results

From the previous step to get a certain degree of overlapping operation frequent sets, retraces the operation of the operation set back to the corresponding Web service, That is, the operation of the operation set is replaced by the Web service which it belongs to, this creates a mashup-related initial Web service frequent set. So now can create two Web services initial frequent set: Service input frequent set, service output frequent set. The final result set of the mashup-related Web service aggregation is generated by the two service frequent sets.

According to the relevant Web mashup aggregation standard of judgment, high similarity of input parameters, that is, the Web services of aggregation in the Web service input frequent set should belong to the same service frequency set; the output parameters of high heterogeneity, that is, the function of the aggregate service different services should be in the Web service output frequently belong on different sets of service sets. If Web services are selected in the same service input frequent set, and these Web services belong to different Web service frequent sets in the service output frequent set, these services form the final Web service related frequent sets to meet Web mashup related aggregation standards.

A Web service can contain several operations, and some of the operations may be the same as other service operations, so the output of Web services with the same input characteristics in a Web mashup aggregation allows some of the same output characteristics. In all Web services in the same service input frequent set a certain percentage of services can appear in the same output service at the same time. When the same feature threshold maxweb is set and the services belonging to the same Web service input frequent set and belonging to the same output frequent set are discarded when the proportion of such services is greater than maxweb, otherwise, the Web service input frequent set is reserved. That is, the services included in the same Web service input frequent set there are at most maxweb services in the same service output frequent set.

3. CONCLUSIONS

This paper is mainly to do the preparation work for Web mashup, and find mashup related Web services which can be integrated. Mashup Related service aggregation is based on the I/O parameters of the WSDL document for the basic processing object, and ultimately to achieve service related aggregation.

4. ACKNOWLEDGMENT

Project supported by the National Nature Science Foundation of China (No.91546110, 61272182, 61100028, 61572117), State Key Program of National Natural Science of China (61332014), Key Laboratory of Computer Network and Information Integration of southeast University (K93-9-2014-03B) and Fundamental Research Funds for the Central Universities (N140404011, N150402002, N150404008).

5. REFERENCES

D. Braga, S. Ceri, F. Daniel, D. Martinenghi, (2008). Mashing Up Search Services, IEEE InternetComputing,12(5):16-23.

IBM QEDWiki

Yahoo!Pipes. http://pipes.yahoo.com

Microsoft Popfly. http://www.popfly.com

programableweb.com

P. Elise, F. Darlene, (2011). Map Mashups and Rise of Amateur Cartographers and Map Makers, ACM LA Bulletin, 127:26-30.

T. S. Dillon, C. Wu, E. CHANG, (2007). GRIDSpece: Semantic grid services on the Web: Evolution towards a SoftGrid, Proceedings of 3rd International Conference on Semantics, Knowledge and Grid. New York: IEEE,:7-13.

K. Gomadam, A. P. SHETH, (2007). SA-REST and mashups: Adding semantics to RESTful services, Proceedings of the International Conference on Semantic Computing, Washington DC, IEEE Computer Society,:469—476.

J. Tatemura, A. Sawires, O. Po, (2007). Mashup feeds: Continuous queries over Web services. Proceedings of the 2007ACM SIGMOD International Conference on Management of Data. New York: ACM, 1128-1130.

Authors

Ying Yin is an associate professor of Computer Science at northeastern university. She received her PHD in 2008 in Computer Science from northeastern university. Her research interests include cloud computing and data mining.

Xizhe ZHANG received a Ph.D. degree in computer application from Jilin University in 2006. Currently, he is an associate professor at Northeastern University. His research interest includes complex network analysis.

Honghong Xu received her MSc in English and Softwar Engineering from DaLian Jiaotong University in July 2006. Currently, he is a graduate student for Master's degree in Northeastern University. His research interest includes machine learning and data mining.

Bin Zhang is a professor of Computer Science at northeastern university. He received his Ph.D. degree from Northeastern University in 1997. His current research interests are cloud computing and data mining.