Constraints based Web Service Semantic Augmentation

Xiaocao Hu, Zhiyong Feng, Shizhan Chen*
Tianjin Key Laboratory of Cognitive Computing and Application
School of Computer Science and Technology, Tianjin University
Tianjin, China
{huxiaocao, zyfeng, shizhan}@tju.edu.cn

***Abstract*—Service relations facilitate the automation of service reuse. Most of studies on the service relations focus on the inputs and outputs. However, different Web Services tend to utilize the same parameters without formally specifying their constraints. Due to this, the semantics, introduced by semantic annotation, is still not rich enough for accurate descriptions, thus generating a large number of inappropriate service relations. To address this, we propose an approach for augmenting semantics of Web Services based on constraints, which can be regarded as a complement to semantic annotation. The semantics is augmented via a hybrid analysis of heterogeneous constraints, including the server constraint and object constraint.**

Keywords-Web Services; Parameter Constraints; Semantic Augmentation

I. INTRODUCTION

With the advent of Web Services, the software industry is evolving from developing specific functionality from scratch to reusing functionalities off the shelf. As the number of Web Services increases, service relations are utilized to facilitate the automation of service reuse.

In general, service relations are identified on the basis of IOPE (input, output, precondition and effect). However, most Web Services may not need any preconditions and have no effects, the focus of service relations are inputs and outputs. Different Web Services tend to utilize the same parameters, and the interpretation of the same parameters may vary depending on the constraints on parameters. However, constraints on parameters are often not formally specified. Thus, the semantics, introduced by the semantic annotation, is still not rich enough for providing more accurate descriptions. As a result, a large number of inappropriate service relations may be generated.

To cope with the problems mentioned above, this paper proposes an approach for augmenting semantics of Web Services based on constraints. The semantic augmentation based on constraints is complementary to semantic annotation, since the combination of semantic annotation and semantic augmentation can describe Web Services more accurately. In particular, we focus on two types of parameter constraints that commonly exist in Web Services, including server constraint and object constraint.

The rest of this paper is organized as follows. Section II reviews the related work. Section III presents the approach through a series of examples. Section IV draws the conclusion.

II. RELATED WORK

In the field of semantic annotation, many solutions [1-2] have been developed. However, there are two issues that commonly exist in these studies. First, these works mainly focus on parameter names. Different Web Services tend to utilize the same parameters without formally specifying their constraints. As a result, parameters with same names but different meanings are mapped to same ontology concepts. Second, these works utilize manually built ontologies as the semantic support. On one hand, building ontologies by hand is time-consuming. On the other hand, the evolution of ontologies is difficulty.

In [3], the utilization of public open ontology is advanced, which fix the second issue mentioned above. Nevertheless, the first issue still exists in [3].

By considering foregoing research works, we believe that the semantics introduced by semantic annotation is still not rich enough for accurate descriptions. This motivates us to augment the constraints semantics of Web Services.

III. SEMANTIC AUGMENTATION BASED ON CONSTRAINTS

As Fig. 1 illustrates, the approach starts with a preparatory analysis, and then proceeds with constraints augmentation.

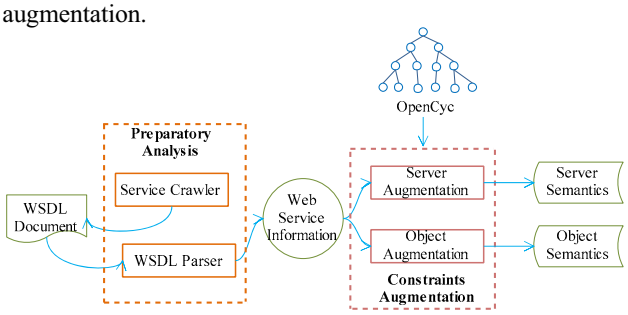

Figure 1. Augmenting semantics of Web Services based on constraints.

A. Preparatory Analysis

The preparatory analysis stage includes two parts, service crawler and WSDL parser [4]. The service crawler fetches WSDL documentations for Web Services from the Internet. For simplicity, the service crawler is simplified by crawling ProgrammableWeb.com. Regular expressions are utilized to extract URLs of WSDL documentations. Only very little manual effort is needed to write the regular expressions,

978-1-4799-5054-6/14 $31.00 © 2014 IEEE
DOI 10.1109/ICWS.2014.109

since pages describing different Web Services in the ProgrammableWeb.com share the same style and structure.

The WSDL parser extracts various types of information needed by the subsequent steps from WSDL documentations, including definitions of service operations, e.g., the available operation names and the list of input and output parameters for each operation, and structural information of parameters, e.g., the user-defined parameters and primitive parameters encapsulated in user-defined parameters.

B. Constraints Augmentation

In particular, we focus on two types of parameter constraints that commonly exist in Web Services, including server constraint and object constraint. The server constraint describes the mechanism for controlling access. For example, when requesting the operation "*createMessage*" provided by the Northtext API, a *username* parameter and a *password* parameter are required to authenticate the user. The valid values of *username* and *password* can only be provided by Northtext. The object constraint describes the target for treatment. For example, when requesting the operation "*ConvertRealTimeValue*" in the XigniteGlobalCurrencies API, a *from* parameter and a *to* parameter are required to specify two currencies for converting. The valid values of *from* and *to* can only be currency symbols.

1) Server Augmentation: We observe that a few patterns are presented in the naming convention of parameters that are related to authentication. For example, if a parameter denotes a valid user name, several most frequently used parameter names would be {username, user_name, userid}. Or if a parameter denotes a license, the possible parameter names would be {licensekey, license_key}. Benefit from this, a parameter name list is predefined to enumerate possible parameter names that may be used to describe parameters related to authentication.

The server augmentation is performed only when a Web Service has one or more parameters contained in the predefined parameter name list. For a parameter related to authentication, its server semantics is the server of the Web Service, which can be extracted from the endpoint. For example, http://admin.northtext.com/api.asmx is endpoint of the Northtext API, in which "http://admin.northtext.com" is the Web Service's server, and "/api.asmx" is the path in the server. Hence, the sever semantics of *username* parameter and *password* parameter is "http://admin.northtext.com".

2) Object Augmentation: Intuitively, the interpretation of a parameter is combined with the context of the parameter, such as the operation along with its description. Nevertheless, most of WSDL files contain little or no textual descriptions, let alone provide descriptions for parameters. Fortunately, almost all primitive parameters are encapsulated in user-defined parameters, which provide the context for primitive parameters, just as Fig. 2 shows. Therefore, user-defined parameters are interpreted as the object semantics of primitive parameters. More specifically, ontology concepts that annotate user-defined parameters are regarded as the object semantics of primitive parameters. The details of annotating user-defined parameters refer to [3]. Note that the basis of object augmentation is that user-defined parameters have meaningful names whose meaning has sufficient discriminative power.

```
▼<s:element name="NearbyAttractionsInfo">
  ▼<s:complexType>
    ▼<s:sequence>
       <s:element minOccurs="1" maxOccurs="1" name="name" type="s:string"/>
       <s:element minOccurs="0" maxOccurs="1" name="category" type="s:string"/>
       <s:element minOccurs="1" maxOccurs="1" name="address" type="s:string"/>
       <s:element minOccurs="0" maxOccurs="unbounded" name="description" type="s:string"/>
     </s:sequence>
   </s:complexType>
 </s:element>
```

Figure 2. An example of the user-defined parameter.

We next use the user-defined parameter shown in Fig. 2 as an example to illustrate the object augmentation. Firstly, NearbyAttractionsInfo is passed to the word segmentation, which generates three words, including nearby, attractions, and info. Since info is short for information, info is replaced by information. Then words nearby and information are filtered out respectively by the POS tagging and stop-word elimination. Subsequently, the stemmer reduces the word attractions to attraction. And then attraction is associated with the concept #$TouristAttraction in the OpenCyc. Hence, TouristAttraction is obtained as the object semantics for the four primitive parameters.

IV. Conclusion

In this paper, we have proposed an approach for augmenting semantics of Web Services based on constraints. The constraints semantics is an effective complement to the semantics of semantic annotation. The semantics is augmented by a hybrid analysis of heterogeneous constraints, including server constraint and object constraint.

From the perspective of semantic annotation, the constraints semantics can provide more accurate descriptions. From the perspective of service composition, the constraints semantics can provide the guarantee of appropriate composite Web Services.

Acknowledgment

This work is supported by the National Natural Science Foundation of China grant 61373035, 61173155 and National High-Tech Research and Development Program of China grant 2013AA013204. Corresponding author: shizhan@tju.edu.cn.

References

[1] A.A. Patil, S.A. Oundhakar, A.P. Sheth, and K. Verma, "METEOR-S Web Service Annotation Framework," Proceedings of the 13th international conference on World Wide Web (WWW '04), ACM, 2004, pp. 553-562, doi:10.1145/988672.988747.

[2] P. Kungas and M. Dumas, "Cost-Effective Semantic Annotation of XML Schemas and Web Service Interfaces," IEEE International Conference on Services Computing (SCC 09), IEEE Computer Society, Sep. 2009, pp. 372-379, doi:10.1109/SCC.2009.17.

[3] X.C. Hu, Z.Y. Feng, and S.Z. Chen, "Augmenting Semantics of Web Services based on Public Open Ontology," IEEE International Conference on Services Computing (SCC 13), IEEE Computer Society, Jun. 2013, pp. 304-311, doi:10.1109/SCC.2013.96.

[4] X.C. Hu, S.Z. Chen, and Z.Y. Feng, "Semi-automatic Acquisition and Formal Representation of OpenAPI," in CCIS, vol. 332, V. Khachidze, T. Wang, S. Siddiqui, V. Liu, S. Cappuccio, and A. Lin, Eds. Heidelberg: Springer, 2012, pp. 86-96.