

Improved Methods for Tagging and Semantic-Annotation for the Semantic-based OpenAPI Retrieval System*

Seung-Jun Cha, Kyu-Chul Lee**

Dept. Of Computer Engineering, Chungnam National University
99 Daehak-ro, Daejeon, Korea
{junii, klee}@cnu.ac.kr

Abstract. The OpenAPI retrieval system has been developed applying semantic web technologies as the typical information retrieval method had. The semantic web technologies applied in the system are tagging and semantic-annotation. However, these methods are difficult for the non-technical general users to use. To solve these problems we suggest improved methods for the tagging and semantic-annotation. Tagging turn human-readable service description to machine-readable one by adding tags to the description. Semantic-annotation means to input related semantic information into parameters. Suggested tagging and semantic-annotation methods are applied and tested through the testbed system. These improvements enabled non-technical general users to applying semantic information to the OpenAPI retrieval system.

Keywords: OpenAPI, Service Retrieval, Tagging, Semantic Annotation

1 Introduction

With the advent of the web, a lot of information and documents are scattered in the web. As user has difficulty in finding information what they want, the information retrieval becomes increasingly important and various retrieval methods are developed.

In the Web 2.0 environment, a lot of service vendors open the API interface of their services; it is named OpenAPI. Throughout the widely use of OpenAPIs, the web is even more expanded as a platform.

According to the website 'ProgrammableWeb', the number of the OpenAPI is enlarged rapidly. For the effective use of OpenAPIs, the retrieval systems were developed [1][2]. The OpenAPI retrieval system has been developed applying semantic web technologies as typical information retrieval method had. The semantic web technologies applied in the system are tagging and semantic-annotation. The methods of tagging and semantic-annotation are defined for adding semantic

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012R1A1A4A01015696).

** Corresponding author.

information to human-readable service descriptions. However, suggested tagging and semantic-annotation are done manually to every service. They are difficult for the non-technical general users to use and the probability of failure is high.

To solve these problems we suggest improved methods for tagging and semantic-annotation. Enhanced tagging method and the semantic-annotation tool are developed.

The rest of the paper is organized as followed. In the section 2 describe about the developed integrated retrieval system for OpenAPI as related work. Enhanced tagging method is shown in the section 3, and developed semantic-annotation tool is shown in the section 4. The section 5 explains about the experimental test results, and conclusions and future works follow in the section 6.

2 Integrated Retrieval System for OpenAPI

In the paper [2], Descriptions of OpenAPI are found and stored in the database using semantic technologies. For applying semantic technologies, the semantic annotated-integrated service information model for OpenAPI (SA-OpenAPI) was defined for adopting various protocols as well as embedding semantic information. Tagging and semantic-annotation methods are also developed to convert the service description into SA-OpenAPI.

2.1 Tagging

Tagging is the method for embedding machine-readable tags into the human-readable service description for extracting necessary metadata automatically. The RDFa [3], the W3C's standard for adding a set of attribute-level extensions to XHTML for embedding rich metadata within web documents, is used for tagging.

In order to apply RDFa standard, RDF schema should be defined beforehand. *BusinessEntity*, *BusinessService*, *Service*, *Port*, and *Operation* is define with the class, and the object property, relation between class, and the data property, the value of the class, is also defined.

Tagging is classified into three types; tagging about the class, tagging about the class and object property, and tagging about the data property. Elements ('div', 'span') and attributes ('src', 'rel', 'about', and 'typeof') are used for tagging complying with the RDFa standard.

Tagging about the class is only used to the root class *BusinessEntity*. The element 'div' is defined for the name of the element. The attribute 'typeof' is for the name of the class and 'about' is for the ID value for the specific class.

2.2 Semantic-Annotation

The semantic-annotation means to annotate semantic related class information of the ontology to the parameters of the service. The paper [4] proposes the SAWSDL [5] based annotation for SOAP-base Web Services and SA-OpenAPI based annotation for RESTful web services and other protocols of service.

The methods of semantic-annotation are to put additional metadata, a *modelReference*, *liftingSchemaMapping*, and *loweringSchemaMapping*, into input/output parameters as an attribute.

A *modelReference* is the reference information of the ontology class having same meaning with a selected input/output parameter. A *liftingSchemaMapping* refer to the conversation information from the datatype of the parameter to the datatype of the ontology class. A *loweringSchemaMapping* is vice versa.

3 Enhanced Tagging Method

The proposed tagging method has a limitation that administrators analyze every HTML page having service description and tag the metadata into them manually. Therefore non-programmers hardly analyze the document as well as tag the metadata. Furthermore it is not guarantee the fault-tolerant as the metadata items are too complex.

To solve these problems, we propose enhanced the RDFa tagging method using the XHTML. XHTML apply the grammatical structure of XML to the HTML [6][7]. By applying the RDFa structure to XHTML, tagging has functions of auto-completion and auto-view. Auto-completion is the function when input specific alphabet, related keywords are automatically completed, and auto-view suggests attribute names related the input element.

To do this, first we modify the DTD for XHTML supporting the attribute names and values defined for the RDF schema, and then examine tools about adopting the modified DTD and supporting auto-completion and auto-view.

3.1 Modification of the HTML DTD

XHTML provides the DTD for structuring. Necessary elements for tagging the RDFa might add in the DTD for having structure of the SA-OpenAPI.

Only the elements 'div' and 'span' are used for tagging. Table 1. shows the modified DTD adding attribute names and values.

Table 1. Modified DTD for XHTML

<pre> <!ELEMENT span %Inline;> <!-- generic language/style container -> <!ATTLIST span property (us:hasBusinessEntityName/us:hasBusinessServiceName/us:hasBusinessServiceDescription/us:hasCategory/us:hasAccesspoint/us:hasPortName/us:hasProtocol/us:hasOperationName/us:hasOperationInput/us:hasOperationOutput/us:hasPortDocumentation/us:hasOperationDocumentation) #IMPLIED> <!ATTLIST span content CDATA #IMPLIED> <!ATTLIST span src CDATA #IMPLIED> <!ATTLIST span about CDATA #IMPLIED> <!ATTLIST span typeof (us:BusinessService/us:Port/us:Operation) #IMPLIED> </pre>
--

3.2 Evaluation

The application XML-SPY is used for evaluating proposed tagging method. XML-SPY basically provides the text editing interface and also support functions for auto-completion and auto-view.

For tagging, first open new XHTML document and apply modified DTD file. Then define the namespace for tagging and semantic-annotation. Three namespaces are defined for using rdfs, tagging, and semantic-annotation.

The function auto-view is used as shown in Fig. 1(a). User want to input 'div' element for defining the class, then input just '<div' text. The possible attribute names 'about', 'rel', 'src', 'typeof' are automatically viewed on the screen. The function auto-completion is used as shown in Fig. 1(b). User want to input relationship about 'BE-01', 'rel=' keyword are input automatically using auto-view. The possible values of the attribute 'rel' are shown on the screen and autocompleted.

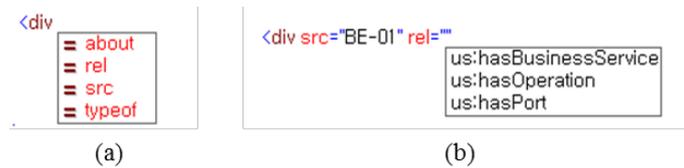


Fig. 1. Results of the Tagging Tool

4. Semantic-Annotation Tool

The semantic-annotation method [4] were suggested manually input semantic related metadata in the service description. It is difficult for non-specialists who never deal with the semantic information or ontology to annotate directly.

To solve this problem, we develop semantic-annotation tool. It simply shows parameter information that required annotation, and provide ontology viewer for effective navigation of the ontology. The tools classified in 4 sections; Operation information, Semantic-annotation, Element information, and Ontology viewer. (Fig. 2.)

4.1 Ontology Viewer

Ontology viewer is designed by analyzing the ontology, and output the relationship of classes in the tree and graphic interface. The OWL API [8] is used for analyzing ontology. As the top class of the ontology is 'Thing', the tree and graphic view expanded from 'Thing'.

The tree viewer provide two functions using just clicking arrow left of the class name or double clicking; Expand Node, Collapse Node. Expand Node expand nodes just one level below from the selected class. Collapse Node collapse sub-nodes from the selected class.

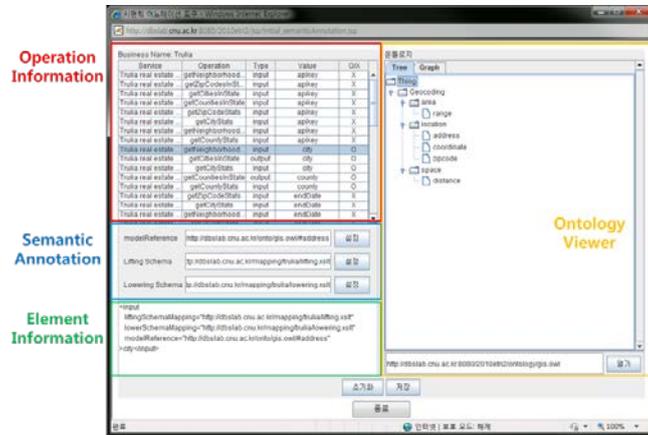


Fig. 2. Semantic Annotation Tool

The essential part of the graphic viewer is based the open source project TouchGraph. The graphic viewer provides additional functions of controlling nodes. *Expand Node*, *Collapse Node*, *Hide Node*, *Center Node*, and *Apply* functions are supported by use of the mouse left button. *Radius*, *Zoom*, *Rotate*, and *Hyperbolic* functions are supported below the tool.

5. Experimental Test

The environments for the experimental test are followed. Fedora release 8 is used for operation system. Systems are working in the Tomcat 5.0 servlet container and MySQL 5.0 database.

5.2 Tagging

Tagging is applied only in the RESTful web service and others. In the interface for Tagging, the source code of HTML for Target service information is stored on the clipboard by using 'Original' button. After do tagging, the document is saved in the file, then uploaded to the system. 'Save' button is reflecting the file to the system. The service description after tagging is shown in Fig. 3.

5.3 Semantic-Annotation

The semantic-annotation can be applied only services which are already tagged. The java applet based semantic-annotation tool is executed and user can do semantic annotation. After annotating, modified service description is reflected in the system. Fig. 3. shows the result after semantic-annotation. The effective for suggest method of tagging and semantic-annotation is verified by the testbed system.

```

</operation>
<name>GetLiveCompactWeatherByCityCode</name>
<input>UnitType</input>
<input>Length</input>
<input>ACode</input>
<input
liftingSchemaMapping="http://dbslab.cnu.ac.kr/mapping/weatherbug/lifting.xslt"
lowerSchemaMapping="http://dbslab.cnu.ac.kr/mapping/weatherbug/lowering.xslt"
modelReference="http://dbslab.cnu.ac.kr:8080/2010etrl2/ontology/gis.owl#zipcode">PostalCode</input>
</input>LiveCompactWeatherData</output>
</operation>
<operation>
<name>GetLiveCompactWeatherByStationID</name>
<input>UnitType</input>
<input>StationID</input>
<input>Length</input>
<input
liftingSchemaMapping="http://dbslab.cnu.ac.kr/mapping/weatherbug/lifting.xslt"
lowerSchemaMapping="http://dbslab.cnu.ac.kr/mapping/weatherbug/lowering.xslt"
modelReference="http://dbslab.cnu.ac.kr:8080/2010etrl2/ontology/gis.owl#zipcode">ACode</input>
</input>LiveCompactWeatherData</output>
</operation>

```

Fig. 3. Service Description after tagging and semantic-annotation

6. Conclusion

The purpose of this paper is to develop enhance methods of tagging and semantic-annotation. Current methods are manual, inefficient, and time-consume. To solve these problems, we studied methods as followed.

Tagging methods are enhanced to input complex defined element and attribute accurately by using suggested auto-completion and auto-view function. Java applet based semantic-annotation tool is also developed. User easily found and adopted the parameter required semantic-annotation.

Developed methods for tagging and semantic-annotation will be integrated other related studies. If related studies, SAWSDL as well as hREST, SA-REST are handled together, various services are utilized integrated.

References

1. Chun, D. S., Cha, S. J., Lee, K. C., Development Integrated Retrieval Methods for OpenAPIs and Mashup Capable Services in u-GIS Environments, Journal of Korea Spatial Information System Society, vol 11/3, pp25-34 (2009)
2. Choi, Y. J., Cha, S. J., Lee, K. C., Development of an Integrated Retrieval System for OpenAPIs Using Semantic Web Technologies, Database Research, vol. 26/3, pp.43-63 (2010)
3. Ben, A., Mark, B., RDFa Primer, Bridging the Human and Data Webs, W3C Working Group Note, <http://www.w3.org/TR/xhtml-rdfa-primer> (2008)
4. Dan, C., Gleaning Resource Descriptions from Dialects of Languages(GRDDL), W3C Recommendation, <http://www.w3.org/TR/2007/REC-grddl-20070911> (2007)
5. Joel, F., Holger, L., Semantic Annotations for WSDL and XML Schema, W3C Recommendation, <http://www.w3.org/TR/sawSDL> (2007)
6. Martin, E. Ken, F. L., Typing XHTML Web Applications in ML, LNCS, vol.305, pp .224-238 (2004)
7. Ben A., Mark, B., Shane, M., Steven, P., RDFa in XHTML:Syntax and Processing, W3C Recommendation, <http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014> (2008)
8. OWL API, <http://owlapi.sourceforge.net>