ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

Student Information Retrieval using RESTdesc

Dr. G. Rama Mohan Babu and Morsa Chaitanya
Professor, Dept. of Information Technology & Asst. Professor, Dept. of Computer Applications
RVR & JC College of Engineering, Guntur, Andhra Pradesh, INDIA

Abstract*— Web service discovery is the process of locating a suitable Web service for a given task. The proposed RESTDesc is a lightweight approach to express Web services functionality by pre- and post-conditions in simple Notation3 rules. This method integrates existing standards and conventions such as HTTP options, Linked headers, and URI templates for discovery and integration of Web services. The proposed approach keeps the complexity to a minimum and enables service descriptions with full semantic expressiveness. It allows to capture the functionality of describe hypermedia API. The sample implementation of multimedia Web services with mock student profiles verifies the effectiveness of this method with the help of EYE Semantic Web reasoner.

***Index Terms*—Semantic Web, Service description, Service discovery, Notation3 rules.**

I. INTRODUCTION

The immense diversity of various multimedia analysis and processing algorithms makes it difficult to integrate them in an automated platform to perform compound tasks. Yet, recent research has indicated the importance of the fusion of different techniques [2]. It is difficult to make different algorithms interoperate if there are no agreements or guidelines on how communication should happen. A coordinating platform can only select algorithms based on their capabilities in presence of a formal description detailing their pre- and post- conditions. The remainder of this paper is organized as follows: Section 2 gives an overview on related work. Section 3 describes RESTdesc approach, Section 4 shows the publicly available student's vocabularies, Section 5 gives conclusion and finally section 6 provides future work.

II. RELATED WORK

A. Web Service Description Language (WSDL)

The first model of the XML-based Web Service Description Language (WSDL, [5, 6]) provides web services description. WSDL focuses on the communicational aspect of Web services, looking from message oriented point of view. The details of the message format are written down in a very longwinded way and concretized to actual binds such as SOAP [12] or plain HTTP [7]. The WSDL description can contain endpoints, which implement the specified bindings.
The two major problems with the WSDL are,
1. The description only provides the mechanisms to characterize the technical implementation of Web services. It does not provide the means to capture the functionality of a service.
2. The description used to generate module source code automatically, which is then compiled into a large problem. If the description changes, the program no longer works, even if such a change leaves the functionality intact.
The WSDL is not well adapted to real-world circumstantial changes. Hence, WSDL cannot offer automatic service discovery at runtime.

B. Semantic Annotations for WSDL (SAWSDL)

The SAWSDL [15] describes a way how to add semantic annotations to various parts of a WSDL document such as interfaces and operations, and input and output message structures. In addition to that, Web services can be assigned a category with the objective of making them discoverable in a central registry of Web services. It also defines an annotation mechanism for specifying the data mapping of XML Schema [9, 20] types to and from ontologies, often referred to as up- and down- lifting. But, SAWSDL inherits all the disadvantages form WSDL, specifically its brittleness and verbosity.

C. REST Services

A REST/RESTful service is built on the following principles [8],

- Servers and clients are separated from each other by a uniform interface. Both server and clients have well-defined responsibilities, also referred to as separation of concerns. This is to guarantee maximum independence from the one and the other.
- All client requests are stateless, this means that each request from a client has all the information that the server needs to process it.
- Responses must define themselves as cacheable or not using standard HTTP caching techniques.
- When layered systems (like load-balancing) are used, this fact must not be exposed to the API user.

In RESTful APIs, resources are identified by URIs [10]. A resource is to be differentiated from its representation. For example, a set of RDF triples (the resource) might be represented in different serializations (syntaxes), such as RDF/XML or Turtle. The manipulation of any of the representations should carry sufficient information to manipulate the original resource. All messages need to be self-descriptive, for example, the media type of a message needs to make clear what can be done with this message. Each representation needs to communicate relevant related resources, or next steps the client can take at each state.

D. Web Application Description Language (WADL)

The WADL [13] is another Web service description format, also XML-based, which does not degrade HTTP to a tunneling mechanism for SOAP, but advocates proper use of all the aspects of the HTTP protocol. While WSDL 2.0 [17] is

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

also capable of specifying bindings to RESTful endpoints, it still requires the abstractions that enable bindings to SOAP and others. The WADL still suffers from the same problem: it does emphasize the technical properties of the underlying service and does not leave any room for the semantics of the task it performs. This also means that there is no way to automatically discover services based on the desired functionality. Joe Gregorio [11] argued that there is no reason why WADL would be used any differently than WSDL.

E. Semantic Markup for Web Services (OWL-S)

The OWL-S [18] is an OWL [19] ontology for describing Semantic Web services in RDF [14]. A service description of OWL-S consists of three parts, a profile, a model and grounding. Some aspects of profile and model are very similar, in the sense that they both describe input, output, preconditions and effects. The difference is that the profile is used for high-level discovery, while the model is used for more detailed condition matching. Finally, the grounding part specifies the implementation of the service, for instance to WSDL, but other groundings are possible (e.g., in SPARQL [22]). The OWL-S input and output types provide more or less the equivalent of what a WSDL message format contains, albeit with RDF types, so there is only a minimal added semantic value on that level. The real possibilities lie in the use of pre- and post- conditions, which allow expressing complex relationships between input and output values, finally capturing the semantics and functionality of the service. The OWL-S offers functional descriptions capable of automatic discovery of the capabilities of a single service, it does not provide mechanisms to express its relation to other services. Also, descriptions contain redundancies and require a fair amount of vocabulary, even to express conceptually simple services, and rely on external groundings for technical implementations.

F. Linked Open Services (LOS)

The requirement to make explicit the relation between input and output is present within the LOS [16] principles. LOS does this by expressing pre- and post- conditions with SPARQL [21] query graph patterns, because RDF currently cannot express quantification. The drawback of this approach is that these patterns also have to be contained inside string literals, like the OWL-S expression languages. This similarly results in the expression of the conditions residing in a different document level from the remainder of the service description.

G. Resource Linking Language (ReLL)

The ReLL [1] aims to provide a natural mapping from RESTful services to RDF. The authors recognize the issues regarding RESTful service descriptions in general and provide an excellent discussion thereof. ReLL differs from this approach in that it only offers “static description of RESTful services that does not cover, new resources or identification and access schemes” [1], whereas this paper specifically aim to address these cases in the context of automated service discovery and consumption.

H. Universal Description Discovery and Integration (UDDI)

The XML-based OASIS standard UDDI [3] was developed to enable the definition of a set of services supporting the discovery and description of (i) businesses, organizations, and other Web service providers, (ii) the Web services that those institutions offer, and finally (iii) the technical interfaces, which may be used to access those services. UDDI was based on a common set of industry standards at that time, including HTTP, XML, XML Schema, and SOAP. The standard was designed to allow for the description and discovery of both public services and non-public in-house services. It was meant to be used as a service broker where parties interested in a special service could go to and retrieve a list of service providers offering the desired service like shipping address verification. Such services would be described in the so-called Green Pages, including not only technical details, but also contact details of the Web service provider.

III. RESTDESC

RESTdesc express algorithm’s functionality in a way that captures its functionality without requiring lengthy specifications. The main goal of RESTdesc is to use existing standards such as the HTTP protocol, Link headers, and URI templates and apply common best practices for implementing multimedia algorithms as true Semantic Web services. In this paper is to provide a versatile description and communication model, enabling fully automated service discovery and execution, even under changing conditions.

@prefix http: <http://www.w3.org/2011/http#>.
{
Preconditions about a certain resource...
}
=> …imply...
{
…That a certain request exists:
_: request http: methodName[…];
http: requestURI[…];
http: resp[…].
This request then effectuates post conditions on the resource.
}.

Listing 1: The RESTdesc description skeleton.

Listing 1 show the general skeleton for RESTdesc descriptions. The expression language used is Notation3 (N3 [4]), which is based on RDF. The essential elements of the above format:

1. The preconditions indicate the state a certain resource should have before being able to take part in the interaction.
2. The post-conditions describe the new state for that resource (or related resource).
3. The request details are exactly what HTTP request should

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

be made to achieve the post conditions.

The above elements are carried together in the form of a rule, which takes care of correct quantification and variable instantiation.

EYE is a backward-forward-backward chaining reasoner design enhanced with Euler path detection. This work is tested on EYE resoner.

IV. STUDENT'S INFORMATION VOCABULARIES

The publicly available vocabularies for student's information such as (college name, school name, CPI, percentage, etc.) then we find the global namespace (URLs) for every term from the swoogle. Swoogle is the semantic web search engine as listed in Table-I. After finding the Global namespace for every term we provide every namespace with identifier so that we don't have to write the whole URL while writing the N3 document. The RESTdesc allows describing to enable access for agents that do not have any prior knowledge. Two related multimedia services introduced here are student face detection and student face recognition. A user agent can upload a photo to the face detection service and use it to check for the existence of faces in the uploaded image. If faces are found, the user agent can use the face recognition service to try to find out more details on the persons whose faces are contained in the image. Each image is considered as a resource represented by a binary image file like /photos/1. Each face is a resource represented by an RDF document serialized in Turtle, or cropped version of the entire image showing only the particular face like /photos/1/faces/1. Each person is a resource like /photos/1/student/1 represented as a string of the student's information. Once faces are detected to follow a link to a Web service that follows for recognition of these faces, or starting from the first student on an image, to follow a link to the next student on the image. The following photo has 15 students and is treated as 15 regions. Each region describes his information. EYE stands for "Euler YAP Engine" and it is a further incremental development of Euler which is an inference engine supporting logic based proofs.

Table I: All the vocabularies (global namespace) that are used in student information N3 documents

	Identifier	Vocabalary
ject,	name, mbox, pastProject, Interest	http://xmlns.com/foaf/0.1/
	Cpi	http://www.burningbird.net/resume_schema#
	Bname	http://xmlns.com/wordnet/1.6
	Graduation	http://ebiquity.umbc.edu/ontology/person.owl
s in Project	TS	http://www.w3.org/2003/01/geo/wgs84_pos
	ID	http://rdfs.org/sioc/ns
	CITY	http://www.w3.org/2000/10/swap/pim/contact
	ContactLocation	http://www.w3.org/2000/10/swap/pim/contact"
	Add	http://www.w3.org/2001/vcard-rdf/3.0

College name	Cname	http://ontoworld.org/wiki/Special:URIResolver/Relation
(12th percentage),	Per	http://www.owlontologies.com/nullontology.owl
10th percentage	Perc	http://example.org/ontologies/VDO
10th School	Scname	http://lsdis.cs.uga.edu/projects/semdis/opus#School
12th School	Sname	http://www.isi.edu/webscripter/bibtex.o.daml
Expertise Area	XA	http://www.owl-ontologies.com/resume.owl
Programming Language	PL	http://owl.mindswap.org/2003/ont/owlweb.rdf
Programming Tools	Tool	http://www.daml.org/tools/tools-ont
Internship	RI	http://pauillac.inria.fr/cdrom_a_graver/ftp/ocomma/com
Mentor	mentorof	http://purl.org/vocab/relationship/
Duration	duration	http://www.aktors.org/ontology/support
Project description	PDESC	http://usefulinc.com/ns/doap
Achievements and Awards	AA	http://xmlns.com/wordnet/1.6/
Positions of Responsibility	PR	http://veggente.berlios.de/ns/RIMOntology
Interests	Hobby	http://www.daml.org/validator/examples/ont2.daml/
Date of Birth	DOB	http://http://www.w3.org/2001/vcard-rdf/3.0/

V. CONCLUSION

The proposed RESTdesc allows automated agents to react on hypermedia links to student's information. This paper explained how services can be described and how these descriptions can be linked together. Essential in this approach is that the client does not need any specific knowledge about the server or even the topic. All this information is discovered at runtime through standard use of the HTTP protocol. The information contained in the RESTdesc descriptions can be extended with ontological and instance knowledge, obtained by dereferencing resource URIs. The proposed approach of link browsing works across different domains and is far more flexible than queries, which have a rigid structure.

VI. FUTURE WORK

Future work will be to prove the applicability of the approach to a broad family of existing RESTful Web services. The application of RESTdesc technologies to different fields and applications, and versatile error handling and recovery based on HTTP status codes. Another interesting area is the collaboration and integration of different services, for example using ontology matching. Furthermore, the development of even more intelligent agents will offer exciting challenges, such as decision optimization when multiple alternative solution paths exist.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

Fig 1. Contains 15 students

REFERENCES

[1] Alarcón R, Wilde E (2010) Linking data from RESTful services. In: Proceedings of the 3rd international workshop on linked data on the Web.

[2] Atrey P, Hossain M, El Saddik A, Kankanhalli MS (2010) Multimodal fusion for multimedia analysis: a survey. In: Multimedia systems. http://www.springerlink.com/index/E31M71152774R630.pdf.

[3] Bellwood T, Capell S, Clement L, Colgrave J, Dovey MJ, Feygin D, Hately A, Kochman R, Macias P, Novotny M, Paolucci M, von Riegen C, Rogers T, Sycara K, Wenzel P, Wu Z (2004) UDDI version 3.0.2.

[4] Berners-Lee T, Connolly D (2011) Notation3 (N3): a readable RDF syntax. W3C Team Submission. http://www.w3.org/TeamSubmission/n3/.

[5] Chinnici R, Moreau J-J, Ryman A, Weerawarana S (2007) Web Services Description Language (WSDL) version 2.0 part 1: core language. W3C Recommendation. http://xml.coverpages.org/ wsdl20000929.html.

[6] Christensen E, Curbera F, Meredith G, Weerawarana S (2000) Web Services Description Language (WSDL) 1.0. http://xml.coverpages.org/wsdl20000929.html.

[7] Fielding R, Gettys J, Mogul J, Frystyk H, Masinter L, Leach P, Berners-Lee T (1999) Hypertext transfer protocol—HTTP/1.1. Request for comments: 2616. http://www.ietf.org/rfc/rfc2616.txt.

[8] Fielding RT, Taylor RN (2002) Principled design of the modern Web architecture. ACM Trans on Int Tech 2(2):115–150. doi:10.1145/514183.514185.

[9] Gao S, Sperberg-McQueen CM, Thompson HS, Mendelsohn N, Beech D, Maloney M (2008) W3C XML Schema Definition Language (XSD) 1.1 part 1: structures. World Wide Web consortium, working draftWD-xmlschema11-1-20080620.

[10] Gonzalez JL, Marcelnez R (2011) Phoenix: fault-tolerant distributed Web storage based on URLs. JoC 2(1):79–86.

[11] Gregorio J (2007) Do we need WADL?, http://bitworking.org/news/193/Do-we-need-WADL.

[12] Gudgin M, Hadley M, Mendelsohn N, Moreau J-J (2007) SOAP version 1.2 part 1: messaging framework, 2nd edn. W3C recommendation. http://www.w3.org/TR/2007/REC-soap12-part1-20070427/.

[13] Hadley M (2009) Web Application Description Language. W3C Member Submission. http://www.w3.org/Submission/wadl/.

[14] Klyne G, Carrol JJ (2004) Resource Description Framework (RDF): concepts and abstract syntax. W3C Recommendation. http://www.w3.org/TR/2004/REC- rdf -concepts-20040210/.

[15] Kopecký J, Vitvar T, Bournez C, Farrell J (2007) SAWSDL: semantic annotations for WSDL and XML Schema. IEEE Internet Comput 11:60–67.

[16] Krummenacher R, Norton B, Marte A (2010) Towards linked open services and processes. In: Proceedings of FIS'2010, pp 68–77.

[17] Lafon Y (2009) Team comment on the "Web Application Description Language" Submission. http://www.w3.org/Submission/2009/03/Comment.

[18] Martin D, Burstein M, Hobbs J, Lassila O (2004) OWL-S: semantic markup for Web services. W3C Member Submission. http://www.w3.org/Submission/ OWL-S/.

[19] McGuinness DL, van Harmelen F (2004) OWL Web ontology language overview.W3C Recommendation. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[20] Peterson D, Gao S, Malhotra A, Sperberg-McQueen CM, Thompson HS (2008) W3C XML Schema Definition Language (XSD) 1.1 part 2: datatypes. In: World Wide Web Consortium, working draft WD-xmlschema11-2-20080620.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 11, May 2015

[21] Prud'hommeaux E, Seaborne A (2008) SPARQL query language for RDF.W3C Recommendation. http://www.w3.org/TR/rdf-sparql-query/.

[22] Verborgh R, Van Deursen D, De Roo J, Mannens E, Van deWalle R (2010) SPARQL endpoints as front-end for multimedia processing algorithms. In: Proceedings of the fourth international workshop on service matchmaking and resource retrieval in the Semantic Web.