

Adopting Web Semantics with Representational State Transfer

Pavithra. K

PG Scholar

Department of Computer Science and Engineering
Coimbatore Institute of Engineering and Technology
Coimbatore, India

Dr. S. Gunasekaran

Prof and Head

Department of Computer Science and Engineering
Coimbatore Institute of Engineering and Technology
Coimbatore, India

Abstract -- Representational State Transfer (ReST) is an abstract model of web architecture. Restful web services are alternative to the traditional SOAP web services in WEB 2.0. The ReST based methodologies paves the way to align semantic data with web architecture. Adding semantics to ReSTful web services helps to overcome the search problem of locating the desired web services. The syntactic and semantic descriptions allow search engines to support a similarity search for ReSTful web services.

The paper intends to elucidate the importance of adding semantics to the ReST architecture by review of the literature.

Keywords—ReST; Web Semantics; RDFa; mash-ups;

I. INTRODUCTION

A web service is a methodology to communicate between two electronic devices over World Wide Web. Basically, there are two types of Web service enabling technologies: SOAP (Simple Object Access Protocol) based and ReST (Representational State Transfer) styled. ReSTful Web services have certain advantages in terms of simplicity, loose coupling, interoperability, scalability and serendipitous reuse. The growing number of ReSTful web services available on the web raises a challenging search problem. Adding semantics help to overcome the challenge. The paper discusses the review on adding semantics to ReST architecture. The forthcoming topics discuss about advantages of adding the semantics to the web using ReST architecture.

II. ANALYSIS OF THE LITERATURE

A. Semantic Web and Web 2.0 with Representational State Transfer (ReST)

The web 2.0 and the semantic web aim at same objective of reuse of data, simple processing and integration. Both web 2.0 and semantic web have its own limitations. The User oriented development has made web 2.0 to evolve into new mashups, but the absence of semantic data limits the complexity that the new mashups can handle. The problem with the semantic web is that, there are pre-defined formats such as OWL and RDF to

depict the data from different sources but no defined access methods to handle the data, which limits the usage of the data. The problems of web 2.0 and the semantic web can be fixed by bridging the two. By combining the web 2.0 and the semantic web, the websites can operate on both the OWL and RDF data and provide access to the multiple clients. The process of combining the web 2.0 and the semantic web is done using Representational State Transfer (ReST) methodology. The method proposed by Robert Battle and Edward Benson [3] used two balancing strategies,

- (i) Semantic bridge for web services uses semantic tags on the SOAP or ReST service
- (ii) Semantic ReST defines a standard way through which the ReST based resources on the web are exposed, modified and queried.

Representational State Transfer is the resource based architecture that operates over HTTP protocol. ReST is the actual standard for the service design in the web application. The resources are identified using URL, which is of the form

REQUEST URL

Protocol://Host/ApplicationPath/ResourceType/ResourceID

The core of the ReST based design is the state transfer operation which is interpreted as the CRUD operation in the web application. One other benefit of the ReST architecture is the type representation can be specified in the request header based on the media type.

The Uniform interface which is one of the important constraints in ReST makes it suitable for web 2.0 applications by organising and accessing data over different medium in a unified way. From the design perspective the ReST approach consists of two basic principles,

- (i) Each user-facing component of an application is modelled as a resource.
- (ii) Each resource in the web application is identified through a URL.

ReST approach also complies with the resource paradigm of the semantic web.

There are large amount of data available on the web through ReST and SOAP services but there is no standard defined to

relate it to the semantic web. By providing such a mark up or standard, the web services can be integrated with the semantic web which helps in data access and semantic query across multiple web sites. The semantic bridge for web services (SBWS) tool was developed to integrate web services with semantic web which wraps around the services defined in WSDL (for SOAP) and WADL (for ReST). For the closer alignment of the web2.0 and semantic web ReST is preferred compared to SOAP as ReST principles are best suited for the web applications. The SBWS tool uses WADL to get the description about the web application and uses custom annotations to define the semantics of the ReST services. WADL and SBWS annotations provide the integration that allows for the semantic web to utilize existing ReST services.

This is different approach to integrate web 2.0 with semantic web, it merges the existing RDF operations with the ReST access points. It involves two parts,

- I. Characteristics of semantic ReST request are defined.
- II. Map HTTP ReST operations on to the semantic world.

By integrating the data modelling of the semantic web with the user participation of web 2.0 the returns will be far greater than either of the two component parts. Thus the paper represents the need for integration between these two worlds and why ReST is ideal to perform the integration.

ReST Syntax

TABLE I. Class-level endpoints for semantic ReST

Operation	HTTP Command
List	GET
Query	GET
Create	POST
Insert	PUT
Remove	DELETE

As an example, an HTTP POST might be sent to the <http://www.example.org/user> endpoint to create a new User resource. The server would create a unique resource handle (user:1000), insert a statement declaring its type (user:1000 a:User), and finally respond with the URI representing that new resource, <http://www.example.org/user/1000>. The client could then interact with this new resource-level endpoint using the following resource-level Semantic ReST conventions TABLE I, represents the class level endpoints for semantic ReST. A client could use the resource-level Semantic ReST conventions to add information with the HTTP PUT operation. The endpoint for this request would be the URI of the new resource just created, <http://www.example.org/user/1000>.

```

PUT
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:swc="http://example.org/example-ontology#"
  xmlns:user="http://example.org/user/">
  <swc:Developer rdf:about="http://www.example.org/user/1000">
  <user:first_name>will</user:first_name>
  <user:last_name>smith</user:last_name>
  </swc:Developer>
  
```

</rdf:RDF>

B. SA-ReST: Services and Mash-Ups

Representational State transfer approach is the light weight implementation of the service oriented architecture. Fig .1, represents the ReSTful services that combine discrete data from different services in to meaningful data sets called the mashups.

Many tools were used to create mashups like Google’s mashup editor, Yahoo’s pipes etc., This facilitates the selection of some number of Restful Web services or other Web resources and chain them together by piping one service’s output into the next service’s input while filtering content and making slight format changes.

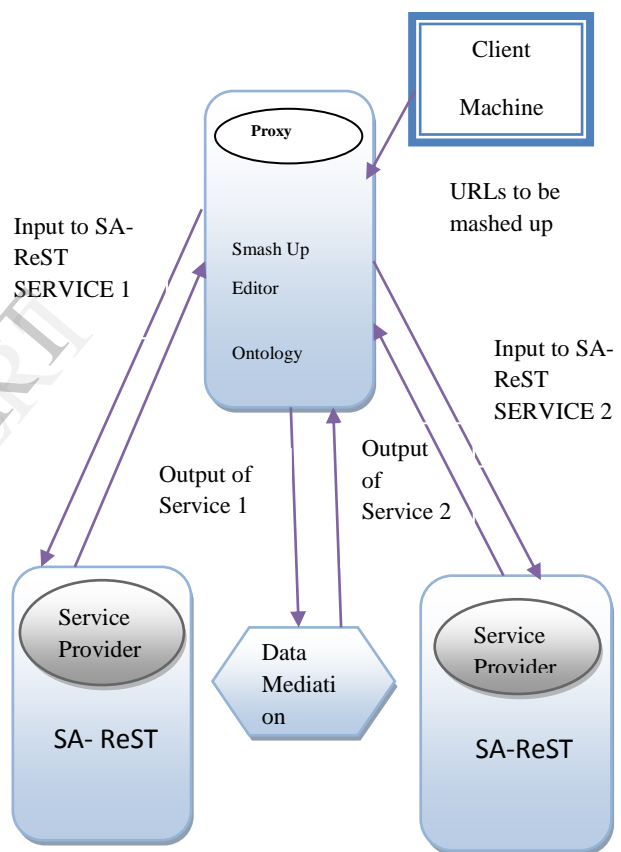


Fig. 1. Service Mashups using SA-ReST

There are certain limitations to the tools like they can interact only with the limited number of services, can interact only with the services that are internal to the organisation and accepts the services that have standard output type like RSS or Atom. These limitations are overcome by adding semantics to the web services. To add semantics to the ReST web service a standard approach SA-ReST was built which is based on the idea of grounding service descriptions to semantic meta models via model reference annotations from SAWSDL. SA-ReST annotations will have to be added to the services that are usually described in web pages composed in HTML. It uses RDFa.

Like SAWSDL, SA-ReST does not force to enforce a language for representing the ontology or the conceptual model. It allows the use of OWL or RDF.

Annotation Techniques and Languages

- (i) SA-ReST uses RDFa which is a subset of RDF, extends XHTML to annotate with markups or annotations, and has built-in support URIs and namespaces, and is recognized by the W3C.
- (ii) To build in greater flexibility and lower the barriers to entry, SA-REST also allows the use of GRDDL for attaching annotations.

Smashups (semantic mashups) are created using semantics to integrate and co-ordinate mashups. Annotations give mashups the ability to know more about a service's inputs and outputs and what the service does, which facilitates data mediation. Amit P. Sheth and Karthik Gomadam [1] stated the importance of adding semantics to ReST methodology facilitates the easy creation of mash ups.

C. Ontologies for ReSTful Services

This focuses to add syntactic and semantic descriptions to WADL file that helps in learning ontology mechanism to enable ReSTful semantic web services [8].

A ReSTful web service is implemented using HTTP and the principles of REST. By using XML-based messaging, ReSTful web services create mashup. A mashup is a web application that combines content from two or more services to create a new service.

The growing number of ReSTful web services available on the web raises the challenging search problem of how to locate the desired web services. Traditional keyword searching is inaccurate, and its limitations have been noted for several years. Adding semantics to ReSTful web services is meant to overcome these limitations.

A specific combination of the existing description language and a learning ontology mechanism in order to enable to develop semantic web services complying with the architectural style ReST was defined. WADL (Web Application Description Language) was used as the language to syntactically describe ReSTful web services. Learning ontology method was addressed to semantically describe the services. It provides both syntactic and semantic descriptions for any given service that allow software agents to automate the discovery and composition of web services.

Adding Semantics to ReSTful Web Services

The absence of description language makes it difficult to achieve the automated discovery and composition of the web services. The focus was on adapting WADL to the syntactic descriptions of a web service. Method of combining WADL and a learning ontology mechanism in order to enable the development of ReSTful semantic web services was proposed. This involves a novel learning ontology method to semantically describe the ReSTful web services which helps

software agents to automate the discovery and composition of the web services.

Automatically generates ontologies from WADLs and their underlying semantics that helps to locate operations of ReSTful web services. To improve the accuracy of the method, the names are pre-processed as follows: (1) after parsing WADL files, if a parameter name contains multiple-words (e.g., ClientName), the name is tokenized, (2) word stemming and stop-word filtering are performed, and (3) after expanding abbreviations, a thesaurus is used to find synonyms. It uses a technique that clusters parameter names in the collection of web services into semantically meaningful concepts.

Benefits of ReSTful Semantic Web Services

The benefits by integrating existing ReSTful web services with semantic ontology are allows a search engine to support similarity searching on the web services, such as finding similar web service operations and finding operations that compose with a given one. The proposed a clustering technique and a pattern analysis technique overcomes limitations of traditional keyword search. The clustering technique gives a breadth of coverage for common terms, while the pattern analysis technique gives a depth of coverage by providing the relationships. By combining these two techniques improve both the recall and the precision of the search. Method of combining syntactic and semantic information gives an agent the ability to know more about a service's inputs/outputs and what the service does, so that the agent can automatically compose web services without human intervention.

D. Use of RDFa in ReSTful Services

This focuses on adding semantics to the services using RDFa. RDFa is compared with other microformat to demonstrate its technical merits and feasibility [7].

In ReST there are two types of state. One is resource state that is about resource information, and the other is application state that is about the path of the client participated in applications. Resource state stays on the server side and application state only lives on the client side.

Service Representations

ReSTful Web Services perform actions on a resource by using a representation to capture the current or intended state of that resource and transferring that representation between server and client. For example: the client manipulates resource state by sending a representation as part of a PUT or POST request. The server manipulates client state by sending representations in response to the client's GET request.

The data format of a representation is known as a media type, the design of a media type can directly impact the user-perceived performance of a system using Web services. The versatility of XHTML makes it can be used to represent all kinds of information. However, this versatility also poses a

problem that is how to understand the information represented by XHTML.

RDFa

RDFa is a lightweight semantic specification released by W3C. RDFa borrowed ideas from microformat and based on RDF. It is committed to become a universal solution that includes random, machine-readable data in the webpage. RDFa embeds structured data and metadata of RDF into webpage representations with markups. RDFa inherits powerful knowledge representation ability of RDF. Anything can be represented by RDF can be added to the webpage by RDFa, and no longer need to use the complicated XML document to describe it. RDFa uses a fixed set of elements and attributes in XHTML, in order to make the webpage handle both human-readable content and the machine-readable complicated semantic information. These semantic markups can be extracted into the RDF triples including subject, object and predicate. They will not affect the page display in the browser.

Representation with RDFa

This illustrates the representation using RDFa with three examples. Triples about an article, triples about a person and triples about a social relationship are taken in to consideration.

Advantages of RDFa

RDFa can represent any resource in triple. The uses of ReSTful Web services have great potential, especially to deal with a variety of data kinds. It is impossible to limit the kind of data. In term of universality, RDFa is the better choice for service representation. RDFa provides a better solution that representing the resources relationship with RDF triple. Extensibility - in RDFa, different source information can be represented with various vocabularies. Standardization - RDFa is a W3C recommendation.

RDFa plays an important role in helping ReSTful Web services to bridge the gap between humans and programs. RDFa helps to improve the self-descriptiveness of ReSTful Web services.

III. RESULT AND DISCUSSION

Fig.2 , elucidates the importance of adding semantics to ReST architecture. ReST is ideal to perform the integration for Web Semantics and Web 2.0. The growing number of ReSTful web services available on the web raises the challenging search problem of how to locate the desired web services. Traditional keyword searching is inaccurate, and its limitations have been noted for several years. Adding semantics to ReSTful web services is meant to overcome these limitations.

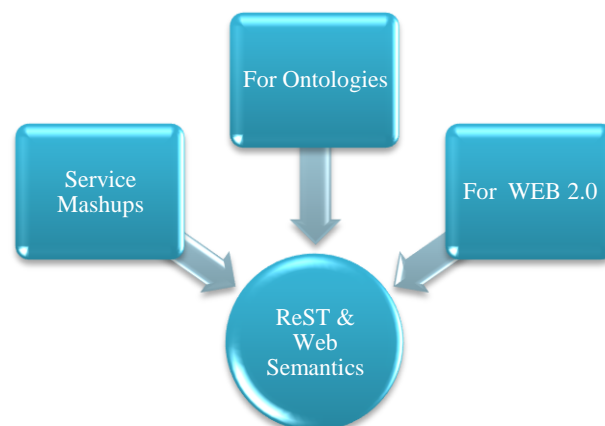


Fig. 2. Use of ReST with Web Semantics

Adding semantics to ReST methodology facilitates the easy creation of mash ups. Semantic mashups are created using semantics to integrate and co-ordinate mashups.

IV. CONCLUSION

Thus paper intends to elucidate the importance of adding semantics to REST architecture which in turn facilitate the creation of service mashups, used for bridging the web semantics with web2.0 and to provide ontology which helps to identify the required web service and by discussing the various contributions towards it.

REFERENCES

- [1] Amit P. Sheth and Karthik Gomadam. (2007) "SA-REST: Semantically Interoperable and Easierto-Use Services and Mashups" Internet Computing, IEEE vol. 11 , no.6.
- [2] Jim Webber, Savas Parastatidis, and Ian Robinson (2010) "REST in Practice" O'Reilly Media.
- [3] Robert Battle and Edward Benson (2008) "Bridging the semantic Web and Web 2.0 with Representational State
- [4] R.T Fielding R.N.Taylor, "Principled design of the modern web architecture", (2002) ACM Trans. Internet Technol. (TOIT) 2 (2) 115–150
- [5] Steve Vinoski (2008) "Demystifying RESTfulData Coupling" Internet Computing, IEEE vol. 12 , no.87-90.
- [6] Transfer (REST)" Journal of web semantics. Volume - 6 Pages 61-69
- [7] Xingyang Feng and Ying Fan (2012) "Research on Application of RDFa in RESTful Web Services" Internet Computing, IEEE
- [8] Yong-Ju Lee and Chang-Su Kim (2010) "Building Semantic Ontologies for RESTful Web Services" Internet Computing, IEEE