2012 Sixth International Conference on Internet Computing for Science and Engineering

Research on Application of RDFa in RESTful Web Services

Xinyang Feng
Henan University of Economics and Law
Zhengzhou Henan, China
hermit2005@sina.com

Ying Fan
Henan Radio & TV University
Zhengzhou Henan, China

***Abstract*—Representational State Transfer (REST) is an architectural style that conforms to Web feature. Recently, people try to design simpler and more scalable Web services based on REST. This kind of Web Services is called RESTful Web Services. Service representation is directly related to the quality of RESTful Web Services. So Representation design is a common-concerned issue. In this paper, we proposed an innovative service representation method with RDFa. Service representations with semantics are developed. RDFa is compared with other microformat to demonstrate its technical merits and feasibility.**

Keywords---RDFa; REST; RESTful Web services; Web services semantics

I. INTRODUCTION

Most of practical Web services are developed based on the Web services protocol stacks, such as Simple Object Access Protocol (SOAP), Web Service Definition Language (WSDL), etc. With the Web services protocols and standards become more mature, there still exits many problems such as a higher cost threshold and potential performance problems in the large-scale concurrent requests [1]. In 2000 the concept and new architecture style called REpresentational State Transfer **(**REST**)** was introduced in the dissertation of Roy Fielding, one of the principal authors who developed the HyperText Transfer Protocol (HTTP). It is not strictly only a method of building what are sometimes called “Web services.” REST has been widely used in IT and in distributed networks. Any web services systems that obey the Fielding’s REST principles are often referred to as “RESTful” systems.

REST is a way to help communicate the basic concepts underlying the Web. To understand REST, it is necessary to understand the definition of resource, representation and state. A resource can be referred to anything entity. A resource can either be a physical object or an abstract concept. In other words, as long as the entity itself is considered as important thing, it can be exposed as a resource. Usually a resource is something that can be stored on a computer and represented as a stream of bits. A representation is any useful information about the state of a resource. Such concept has been used in the implementation of Web services and Web applications for establishing networks of things. A resource may have multiple different representations. 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. REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems[2]. Some key constraints in REST are as follows:

- Everything being resource;
- Resource identification through URI;
- Uniform interface;
- Manipulation of resources through representations;
- Self-descriptive messages;
- Stateless interactions;
- Hypermedia as the engine of application state.

Web is the most distinguished distributed system using REST. REST conforms to Web feature naturally. Compared to Remote Procedural Call (RPC), REST is more simple and scalable.

II. RESTFUL WEB SERVICES

Web services adopting REST are called RESTful Web services [3]. RESTful Web services recently become popular technical topics and technology. RESTful Web services utilize HTTP standard operations that GET, POST, PUT and DELETE to request and respond. This way means all RESTful Web Services have the uniform interface. In RESTful Web services, resources are identified with URL, and client and server can exchange resource representations contain forms and links to make application states change.

RESTful Web services have many advantages compared to Web services using SOAP, such as architecture simplicity and high scalability. However, RESTful Web service as a new technology lacks in a unified service representation format. In this context, how to find a simple and efficient method of RESTful Web services representation formatting becomes very important.

III. SERVICE REPRESENTATIONS

Representation is any useful information about the current state of resources. Using RESTful Web services, client obtains not resources but representations of resources. A representation consists of data, metadata describing the data, and, on occasion, metadata to describe the metadata (usually for the purpose of verifying message integrity). Metadata is in the form of name-value pairs, where the name corresponds to a standard that defines the value's structure and semantics.

978-0-7695-4705-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ICICSE.2012.44

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. This is where the name “Representation State Transfer” comes from.

The data format of a representation is known as a media type [4]. Some media types are intended for automated processing, while others are intended to be rendered for viewing by a user. A few of them are capable of both. The design of a media type can directly impact the user-perceived performance of a system using Web services. Any data that must be received before the recipient can begin rendering the representation adds to the latency of an interaction. A data format that places the most important rendering information up front (such that the initial information can be incrementally rendered while the rest of the information is being received) results in much better user-perceived performance than a data format that must be entirely received before rendering can begin. For example, a Web browser that can incrementally render a large HTML document while it is being received provides significantly better user-perceived performance than one that waits until the entire document is completely received prior to rendering, even though the network performance is the same.

Representation is actually serialized data structures. Each data structure is good at expression of a particular resource data. An appropriate representation format is important to express resource information conveniently and accurately. XHTML is the abbreviation of Extensible Hypertext Markup Language. Its media type is application/xhtml+xml[5]. By the end of 2000, the W3C released XHTML 1.0 version. XHTML 1.0 is a new language based on the optimization and improvement of HTML 4.0. It aimed at constructing XML-based applications. XHTML introduces the scalability of XML into HTML, while keeping the direct viewing of HTML. XHTML is good at displaying data and describing data structure. Using XHTML as the representation format can not only transfer the resource state, but also provide the lever of resource state in the form of link. Therefore, XHTML is the most common representation format currently.

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. Such that the following XHTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN "
"http://www.w3.org/TR/XHTML11/DTD/xhtml11.dtd ">
<html xmlns= "http://www.w3.org/1999/xhtml" xml:lang= "en">
<head>
<title>An Article Representation</title>
</head>
<body>
<div>
  <h2>Representation State Transfer</h2>
  <h3>Roy T. Fielding</h3>
  ……
</div>
</body>
</html>
```

Figure 1. The XHTML representation

Figure 1 is a representation of an article. The title of the article is in <h2>. The name of the author is in <h3>. Undoubtedly, this representation can be parsed and display by browser. But the information that Representation State Transfer is the title of the article and Roy T. Fielding is the author of the article's can not be communicated to the computer. If the internal data structures in representation can be understood by the computer, then the computer will be able to do more business processing. For example: the same author's articles are classified and so on. To make these data information in XHTML representation understood only by human can also be understood by computer programs, semantics should be add into XHTML representation.

IV. RDFa

RDFa is a lightweight semantic specification released by W3C. RDFa borrowed ideas from microformat and based on RDF[6]. 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.

V. Representation with RDFa

RDFa uses XHTML1.1 attributes. In addition, RDFa can also be used inside HTML4. Three examples of representations with RDFa are introduced hereby. In order to avoid our emphasis being covered by complex resource information, the representations are simplified enough to illustrate how RDFa works.

A. Representation of an Article

Article is a common network resource. The representations of articles often appear in blogs and bulletin board system. The following is a representation of an article with RDFa markups, shown in Figure 2.

```
<div xmlns:dc="http://purl.org/dc/elements/1.1/">
<div about="http://www.sample.com/ article/restful-services">
<h2 property="dc:title"> Representation State Transfer </h2>
<h3 property="dc:creator"> Roy T. Fielding </h3>
</div>
</div>
```

Figure 2. Representation of an article

Figure 2 expresses the following information. Assume there is an article resource at URL “http://www.sample.com/ article/restful-services,” the title of the article is “Representation State Transfer”, and the author is “Roy T. Fielding”. Figure 2 contains two assertions, “about” attribute indicates the subject, and “property” attribute indicates predicate, object in Figure 2 is a literal string.

Dublin Core is a catalogue model produced by Dublin Core Metadata Initiative. It is widely used to describe digital materials such as video, sound, image, text, and composite media like web pages [7]. Dublin Core vocabulary is imported into Figure 2. The prefix “dc” stands for namespace “http://purl.org/dc/elements/1.1/”, thus “dc:creator” and “dc:title” are short-hands for “http://purl.org/dc/elements/1.1/creator” and “http://purl.org/dc/elements/1.1/title”. The two elements have been defined in Dublin Core metadata element set. The elements stand for the author and title of work respectively. Two assertions have same subject. The relationship is shown in Figure 3.

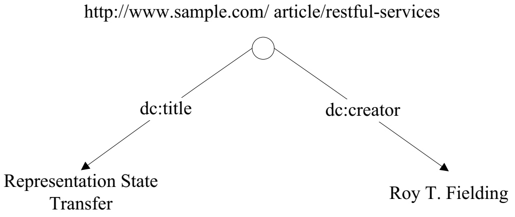
http://www.sample.com/ article/restful-services
dc:title
dc:creator
Representation State Transfer
Roy T. Fielding

Figure 3. Triples about an article

B. Representation of a Person

Here another representation of a person is provided. In order explain the mechanism of the representation. We introduce a concept of FOAF. FOAF is the abbreviation of “Friend of a Friend”. It is a vocabulary for describing, in computer-readable form, the sort of personal information that you might normally put on a home webpage [8]. FOAF can be used to describe person and their basic characteristics, such as name, email address, etc. The FOAF vocabulary is divided into several categories: “FOAF Basics”, “Personal Info”, “Online Accounts / IM”, “Projects and Groups”, and “Documents and Images”. Resource information about person entity is designed with FOAF Basics. The information include name, email address and phone number three items. Each item can be expressed by a RDF triple.

```
<div typeof="foaf:Person"
xmlns:foaf="http://xmlns.com/foaf/0.1/">
<p property="foaf:name">
Alice
</p>
<p>
Email: <a rel="foaf:mbox" href="mailto:Alice@gmail.com">
Alice@gmail.com
</a>
</p>
<p>
Phone: <a rel="foaf:phone"
href="tel:+86-0371-63519883">63519883</a>
</p>
</div>
```

Figure 4. Representation of a person

Different from the Figure 2, Figure 4 does not explicitly specify subject with “about” attribute. The subject is specified implicitly with “typeof” attribute instead, and the type of subject is “foaf:Person”. The first assertion states person name. Its predicate is specified with “property” attribute, and the object is a literal string. Unlike the first assertion, objects of the other two assertions are not literal strings but URLs. So instead of using “property” attribute, the “href” attribute is used to specify a subject. The predicate is specified with “rel” attribute. Figure 5 shows the structure of the triples.

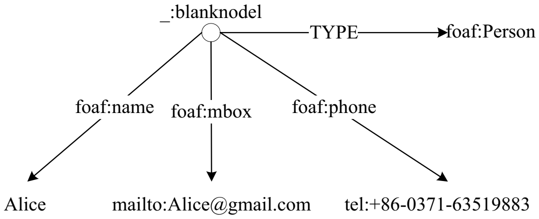

Figure 5. Triples about a person

In Figure 5, the name, email address, and phone number are associated with a blank node. Blank node has no URL to identify it. Instead, many of them have a “typeof” attribute that identifies the type of data they represent. This approach—providing no name but adding a type— is particularly useful when subject is anonymous.

C. Representation of social relationships

RDFa can use the metadata vocabulary to describe the variety of relationships. In the following, the social relationships between people are represented using RDFa. In the representation of social relationships the “Personal Info” in FOAF will be used.

```
<div xmlns:foaf="http://xmlns.com/foaf/0.1/"
about="http://www.sample.com/person/Alice" rel="foaf:knows">
<ul>
<li typeof="foaf:Person">
<p property="foaf:name">Bob</p>
<p>Email: <a rel="foaf:mbox" href="mailto: Bob @sina.com">
Bob@sina.com </a></p>
<p>Phone: <a rel="foaf:phone"
href="tel:+86-0371-63519881">63519881</a></p>
</li>
<li typeof="foaf:Person">
<p property="foaf:name">Mike</p>
<p>Email: <a rel="foaf:mbox" href="mailto: Mike @tom.com">
Mike @tom.com </a></p>
<p>Phone: <a rel="foaf:phone"
href="tel:+86-0371-63519882">63519882</a></p>
</li>
<li typeof="foaf:Person">
<p property="foaf:name">Rose</p>
<p>Email: <a rel="foaf:mbox" href="mailto: Rose @sohu.com">
Rose @sohu.com</a></p>
<p>Phone: <a rel="foaf:phone"
href="tel:+86-0371-63519885">63519885</a></p>
</li>
</ul>
</div>
```

Figure 6. The representation of social relationships

The Figure 6 indicates that the information that Alice knows Bob, Mike and Rose. The “about” indicates that a subject can be given as “http: //www.sample.com /person/Alice”. That is a personal resource. The predicate is “rel =’foaf: knows’”, “foaf: knows” express such a social relationship that subject “knows” object. There are three objects implicitly declared by the “typeof” attribute, Their type are “foaf:Person”. This means that three new blank nodes with their own unique properties will be created. These blank nodes do not need url, and their properties are described by FOAF. Relationships are illustrated as shown in Figure 7.

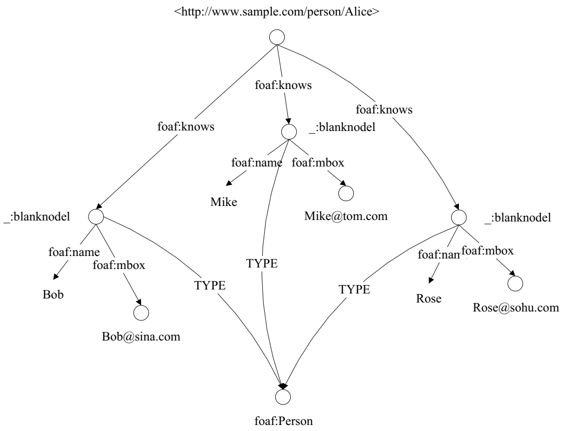

Figure 7. Triples about social relationships

VI. ADVANTAGES OF RDFA

RDFa is not the only metadata annotation specification. There are many other kinds of microformat, Such as hCalendar, hCard and hAtom etc. hCalendar is a specification of representing events on a calendar or planner. hCard is a specification of representing contact information for people and organizations. hAtom is a specification of representing news or blogs. All these microformats have their own features. They better representing the information in respective domain. However, if they are applied to scenarios as general in services representations, they may have disadvantages. Compared to microformats, RDFa has merits in many aspects.

Generality - the microformats, such as hCalendar, hAtom, etc, focus on a certain type of semantics. They are not applicable outside the specific field. RDFa inherits the representing capability of RDF. Thus 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.

Representing relationship - microformats can represent resource information in certain domain, but they are not for representing the resources relationship. This is a shortcoming for microformats as services 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, such as Dublin Core, FOAF, etc. When a new kind of resource information arises, all we have to do is importing the relevant vocabulary. The syntax of RDFa needs no modification. Microformats don't have the flexibility. We have to create a new type of microformat for the new resource. Therefore, microformats have lower extensibility compared to RDFa.

Standardization - RDFa is a W3C recommendation. It is generally accepted by developers. RDF is the basis for the W3C Semantic Web project. Currently there have been some applications supporting RDF triples. RDFa also uses RDF triple to represent semantics. Taking RDfa as the service representation format can leverage existing applications to reduce cost.

VII. CONCLUSION

Using XHTML with RDFa can form service representations with semantic information. RDFa helps to improve the self-descriptiveness of RESTful Web services. RESTful web services using RDFa will gradually be recognized by more RDFa-aware software. When the human-readable data information in service representations can be understood by program, the RESTful Web services would be more helpful. RDFa plays an important role in helping RESTful Web services to bridge the gap between humans and programs.

PEFERENCE

[1] X. Feng, J. Shen, Y. Fan, "REST: An alternative to RPC for Web services architecture," *Future Information Networks*, Oct, 2009.

[2] R. T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," Doctorial Dissertation, Dept. of Computer Science, Univ. of California, Irvine, 2000.

[3] L. Richardson, S. Ruby, "RESTful Web Services," O'Reilly Media, 2007.

[4] Freed N, Borenstein N, "Multipurpose internet mail extensions (MIME) part one: Format of internet message bodies," http://www.ietf.org/rfc/rfc2045.txt, 1996.

[5] S. Pemberton, M. Altheim, D. Austin, "XHTML™ 1.0: The Extensible HyperText Markup Language," http://www.w3.org/TR/2000/REC-xhtml1-20000126/, 2000.

[6] B. Adida, M. Birbeck, "RDFa primer 1.0: Embedding rdf in XHTML," W3C working draft, Dec, 2007.

[7] D. Hillmann, "Using dublin core," http://dublincore.org/documents/usageguide/, 2005.

[8] D. Brickley, L. Miller, "FOAF vocabulary specification," http://xmlns.com/foaf/0.1/, 2005.